



OULUN YLIOPISTO
UNIVERSITY of OULU

KARVONEN VESA-PEKKA & LAUKKA PASI
SUOMALAISTEN OPETTAJIEN ASENTEITA JA VALMIUKSIA OHJELMOIN-
NIN OPETUKSEEN

Kasvatustieteen pro gradu -tutkielma

KASVATUSTIETEIDEN TIEDEKUNTA

Kasvatustieteiden ja opettajankoulutuksen yksikkö
Teknologiapainotteinen luokanopettajakoulutus

2016



Kasvatustieteiden tiedekunta

Tiivistelmä opinnäytetyöstä

Luokanopettajankoulutus		Tekijät Karvonen Vesa-Pekka & Laukka Pasi	
Työn nimi Suomalaisten opettajien asenteita ja valmiuksia ohjelmoinnin opetukseen			
Pääaine Kasvatustiede	Työn laji Pro gradu	Aika Tammikuu 2016	Sivumäärä 58 sivua + 1 liite (6 sivua)
Tiivistelmä <p>Tässä pro gradu -tutkielmassa tarkastellaan suomalaisten perusopetuksen opettajien valmiuksia opettaa ohjelmointia sekä asenteita ohjelmoinnin opettamista kohtaan. Lähtökohtana tutkielmalle on uusi <i>Perusopetuksen opetussuunnitelman perusteet</i> ja sen mukanaan tuoma ohjelmoinnin opetuksen sisällyttäminen opetukseen. Pro gradun teoreettinen osa ja sen pohjalta laadittu mittaristo perustuu kahteen pääteoriaan. Tutkielmassa kerätty aineisto on avointa ja se on kaikkien käytettävissä Aalto-yliopiston kautta. Aineistonkeruun instrumentissa opettajien teknologia-asenteen taustalla ovat Davisin (1989) teknologian hyväksymismalli (Technology Acceptance Model, TAM) sekä Venkatesh et al. (2003) Yhdistetty teoria teknologian hyväksymisestä ja käytöstä (Unified Theory of Acceptance and Use of Technology, UTAUT). Tutkielmassa on tarkoituksena tarkastella suomalaisen perusopetuksen opettajan valmiuksia opettaa ohjelmointia sekä asenteita niin ohjelmoinnin asemasta perusopetuksessa, kuin ohjelmoinnin opetuksesta yleisesti. Työn tuloksien avulla pyritään kehittämään suomalaisten peruskoulun opettajien koulutusta ohjelmoinnin opetuksessa. Pro gradu tutkielman aineisto on kerätty yhteistyössä koodiaapinen.fi-sivuston ohjelmointikurssin kanssa. Tutkimuskysely järjestettiin kurssin osallistujille. Tutkielma perustuu 317 suomalaisen opettajan web-kyselylomakkeeseen.</p> <p>Tutkielman päätulokset: Noin yksi kymmenestä vastaajasta ilmoitti pystyvänsä opettamaan joitakin seuraavista kielistä: Scratch, Scratch Jr., Code.org, ja HTML/CSS. Muiden ohjelmointikielten ja -ympäristöjen osajia oli merkittävästi vähemmän. Uskoa ohjelmoinnin opettamisen osaamiseen vastaajien keskuudessa kuitenkin on, sillä lähes puolet vastaajista ilmoitti pystyvänsä ratkaisemaan ohjelmoinnin opetuksen ongelmia, vaikka paikalla ei olisi ketään ohjeistamassa ja yli 70 % vastaajasta ilmoitti pystyvänsä ratkaisemaan ohjelmoinnin opetuksen ongelmia, jos käytettävissä on runsaasti (riittävästi) aikaa.</p> <p>Ohjelmoinnin opettaminen peruskoulussa nähdään tarpeellisena vastaajien keskuudessa. 74,1 % on osittain samaa mieltä tai täysin samaa mieltä väittämän ”Ohjelmoinnin opettamista tarvitaan peruskoulussa”. Suomalaisten opettajien suhtautuminen ohjelmoinnin opettamiseen on hieman neutraalia positiivisempi tai opettajat eivät tiedä miten he asennoituvat ohjelmointia kohtaan. Yleinen suhtautuminen ja ahdistuksen vähyys kertovat kuitenkin opettajien ottavan ohjelmoinnin opettamisen ennakkoluulottomasti käyttöönsä.</p>			
Asiasanat Teknologian hyväksymismalli, TAM, ohjelmointivalmiudet, asenne ohjelmointia kohtaan, perusopetus, opettajat			



Luokanopettajankoulutus		Authors Karvonen Vesa-Pekka & Laukka Pasi	
Title of thesis Suomalaisten opettajien asenteita ja valmiuksia ohjelmoinnin opetukseen			
Major subject Kasvatustiede	Type of thesis Pro gradu	Year January 2016	No. of pages 58 + 1 appendix (6 pages)
Abstract <p>The main objective of this Master’s thesis is to explore Finnish primary school teachers’ readiness to teach programming and their attitudes towards teaching of programming. The new National Curriculum of Finland requires teachers to be able to teach programming commencing from autumn 2016. Koodiaapinen.fi has designed a massive open online course to educate Finnish teachers on teaching of programming to tackle this requirement. The research questionnaire was offered to the teachers who participated in the course. The results of this thesis are intended to be used on developing of the Finnish primary school teacher training on the area of teaching of programming.</p> <p>Research method of this thesis was web-based questionnaire (n=317). Instrument of the research is based on TAM (Technology Acceptance Model) (Davis, 1989.) and UTAUT (Unified Theory of Acceptance and Use of Technology) (Venkatesh et al., 2003.). Those models were translated and modified to match the research problem and the language of this thesis. One great contributor to the instrument was the questionnaire developed by the researchers of ”<i>The programming studio: investigating teachers’ readiness for teaching programming in the Island of Ireland</i>”. The questionnaire used in this thesis is partially similar to the one used in Ireland to allow direct comparison of the results between these studies.</p> <p>Main results of the thesis: About one in ten of the respondents declared to be able to teach some of the following programming languages: Scratch, Scratch Jr., Code.org and HTML/CSS. The respondents were significantly less familiar with other programming languages. However, almost half of the respondents indicated ability to solve problems of the teaching of programming even if there were no help available and over 70 % of the respondents declared they would solve those problems if they had enough time to work on the problem. Teaching of programming was seen necessary among the respondents as 74,1 % partially agreed or strongly agreed with the statement “<i>the teaching of programming is needed in primary school</i>”. The attitude towards teaching of programming is somewhat more positive than neutral among the respondents or the respondents were not yet able to state their attitudes. The general attitude and lack of anxiety among the respondents denotes that Finnish teachers are open-mindedly welcoming programming into primary school’s curriculum.</p>			
Keywords Technology acceptance model, TAM, readiness for teaching programming, attitudes towards teaching of programming, primary school, teachers			

Sisälllys

TAULUKOT

KUVIOT

LYHENTEET JA KÄSITTEET

1. Johdanto	1
1.1. Ohjelmoinnin opetus julkisessa keskustelussa	2
1.2. Tutkimuksen tavoitteet, rajausta ja rakenne	3
2. Ohjelmointi.....	5
2.1. Mitä on ohjelmointi?	5
2.1.1. Tietokoneohjelma ja konekieli	6
2.1.2. Ohjelmointikieli	7
2.2. Näkökulmia ohjelmoinnin opetukseen	8
2.2.1. Koodauksen opetus on ajattelun opettamista	10
2.2.2. Ohjelmoinnin opetuksen lähestymistapojen tarkastelua.....	11
2.2.3. Yhteenveto	13
3. Tutkimuksen viitekehys	14
3.1. Teknologian hyväksymismalli	14
3.2. Yhdistetty teoria teknologian hyväksymisestä ja käytöstä	16
3.3. Muuttujien kuvaus	18
3.4. Aiempi tutkimus aiheesta	20
3.5. Tutkimuksen konteksti: Ohjelmointi opetussuunnitelmassa	21
4. Tutkimuksen toteutus.....	23
4.1. Deskriptiivinen määrällinen tutkimus tutkimusstrategiana	23
4.2. Verkkokysely aineistonkeruumenetelmänä	24
4.3. Aineiston keruu	25
4.4. Aineiston käsittely.....	26
4.5. Aineiston analyysi	27
4.6. Summamuuttujien rakentaminen.....	29
4.7. Tutkimuksen luotettavuuden arvioinnin kriteerit	31
4.8. Tutkimuksen eettisyyden arvioinnin kriteerit.....	32
5. Tutkimustulokset	34
5.1. Vastaajista saadut tiedot.....	34
5.2. Ohjelmointiosaaminen ja -kokemus	37
5.3. Näkemys ohjelmoinnin asemasta perusopetuksessa.....	39

5.4.	Asenteet ohjelmointia kohtaan	40
6.	Tutkimustulosten tarkastelu	44
6.1.	Vastaajista saadut tiedot	44
6.2.	Ohjelmointiosaaminen ja -kokemus	45
6.3.	Näkemys ohjelmoinnin asemasta perusopetuksessa	47
6.4.	Asenteet ohjelmointia kohtaan	48
7.	Yhteenveto ja pohdinta	53
7.1.	Keskeiset tutkimustulokset	54
7.2.	Tutkimuksen rajoitukset sekä luotettavuuden ja eettisyyden loppuarviointi	55
7.3.	Tutkimuksen hyöty ja jatkotutkimuksen tarpeet	57
	Lähteet.....	59
	Liitteet	64

TAULUKOT

Taulukko 1: Summamuuttujien rakentaminen.....	30
Taulukko 2: Vastaajien sukupuoli	34
Taulukko 3: Vastaajien ikäjakaumat	35
Taulukko 4: Vastaajien opettama luokka-aste	36
Taulukko 5: Käytön vapaaehtoisuus.....	36
Taulukko 6: Vastaajien aiempi ohjelmointikoulutus	37
Taulukko 7: Vastaajien ohjelmointiosaaminen	38
Taulukko 8: Opettajien ongelmanratkaisukyky	39
Taulukko 9: Ohjelmoinnin opettamista tarvitaan peruskoulussa.....	39
Taulukko 10: Työpaikallani suhtaudutaan positiivisesti ohjelmoinnin opettamiseen	40
Taulukko 11: Summamuuttujien keskiarvot (N=317).....	41
Taulukko 12: Summamuuttujien korrelaatiot (N=317)	42
Taulukko 13: Ohjelmointikoulutuksen vaikutus summamuuttujiin	43

KUVIOT

Kuvio 1: Perusmalli teknologian omaksumisesta yksilötasolla.....	14
Kuvio 2: Teknologian hyväksymismalli (TAM)	16
Kuvio 3: Keskeisten muuttujien keskiarvot.....	50
Kuvio 4: Pearsonin korrelaatiot muuttujien välillä.....	51
Kuvio 5: Ohjelmointikokemuksen vaikutus summamuuttujiin	52

KESKEISET KÄSITTEET JA LYHENTEET

A	Attitude (Asenne ohjelmoinnin opettamista kohtaan)
AX	Anxiety (Ahdistuneisuus)
BI	Behavioral Intention (Aikomus opettaa ohjelmointia)
PEU	Perceived Ease of Use (Koettu helppokäyttöisyys)
PUT	Perceived Usefulness to Teachers (Koettu hyödyllisyys)
SN	Subjective Norm (Sosiaalinen vaikutus)
TAM	Technology Acceptance Model (Teknologian hyväksymismalli)
UTAUT	Unified Theory of Acceptance and Use of Technology (Yhdistetty teoria teknologian hyväksymisestä ja käytöstä)

1. Johdanto

Tämän kasvatustieteen pro gradu tutkielman lähtökohdat ovat Opetushallituksen 22.12.2014 hyväksymässä vuoden 2014 *Perusopetuksen opetussuunnitelman perusteissa*, jonka yksi merkittävistä eroista nykyiseen opetussuunnitelman perusteisiin on ohjelmoinnin ottaminen osaksi alakoulun perusopetusta. Käytännössä OPS2014 edellyttää jokaisen opettajan kykenevän opettamaan ohjelmointia jo syksyllä 2016 (Opetushallitus, 2014a, 1). Tutkielman lähtölaukauksena toimi koodiaapinen.fi-sivuston suomalaisille opettajille järjestämä ohjelmointikurssi, josta tarjottiin mahdollisuutta pro gradu tutkielman tekemiseen. Kurssi on *massive open online course* (MOOC) ja sen alustana toimii Eliademy-sivusto. Kurssi järjestetään pääasiallisesti viiden henkilön talkoovoimin, koordinaattorina toimii Aalto-yliopiston oppimispsykologi. Kurssille on ilmoittautunut kurssin aloittamisajankohtana 2572 suomalaista opettajaa (Koodiaapinen, 2015). Syynä kurssin järjestämiselle on OPS2014 sisältämät muutokset muun muassa matematiikan oppisisältöihin sekä tästä ilmennyt akuutti tarve kouluttaa opettajia opettamaan ohjelmointia. Ohjelmoinnin sisällöt on jaettu *Perusopetuksen opetussuunnitelman perusteissa* luokka-asteittain siten, että vuosiluokilla 1—2 oppilaat tutustuvat ohjelmoinnin alkeisiin laatimalla vaiheittaisia toimintaohjeita, joita myös testataan. Vuosiluokilla 3—6 oppilaat suunnittelevat ja toteuttavat ohjelmia graafisessa toimintaympäristössä ja vuosiluokilla 7—9 edetään vaiheittain kohti algoritmisen ajattelun kehittämistä ja ohjelmoinnin käyttämistä ongelmien ratkaisemiseen (Opetushallitus, 2014a, 129, 235, 375).

Tutkielmassa on tehty yhteistyötä Irlantilaisen tutkimuksen ”*The programming studio: investigating teachers’ readiness for teaching programming in the Island of Ireland*” tekijöiden kanssa. Tutkimuksen tekijät luovuttivat kehittämänsä mittarin tämän tutkielman käyttöön, joten tutkimuksien kyselylomakkeet ja analyysimenetelmät ovat keskeisiltä osiltaan samankaltaisia. Tämä mahdollistaa tulosten vertailun, tosin tietyin rajoituksin.

Tutkielman kirjoittajat ovat teknologiapainotteisen luokanopettajakoulutuksen opiskelijoita Oulun yliopistossa. Teknologiapainotteisen luokanopettajakoulutuksen yksi keskeisimpiä asioita on ongelmanratkaisutaitojen kehittäminen, johon ohjelmointi soveltuu myös kirjoittajille muodostuneen näkemyksen mukaisesti erinomaisesti (Lindh, 2014, 6). Ohjelmointi

ei kuitenkaan vielä tätä kirjoitettaessa ole kovinkaan vahvasti esillä luokanopettajakoulutuksen perusopinnoissa.

1.1. Ohjelmoinnin opetus julkisessa keskustelussa

Ohjelmoinnin opetuksen tarvetta on perusteltu useiden tahojen toimista. Muun muassa julkaisu *Opetusministeriön työryhmämuistioita ja selvityksiä 2007:23* sisältää *Tarve laskennallisen tieteen osaamiselle* -kappaleessa maininnan ohjelmointiin ja ohjelmistoihin liittyvien osaamistarpeiden kasvamisesta ja kytkeytymisestä menetelmäosaamiseen. Selvityksen mukaan yliopistokoulutuksen ja -tutkimuksen tulisi tuottaa riittävästi laskennallisen tieteen osaajia, sillä osaamistarpeet kehittyvät nopeasti. Taustalla vaikuttaa teollisuuden huoli tulevaisuuden koodaustaitoisten riittämättömyydestä (Heikkinen, Haataja & Halonen, 2007, 16). Opetusministeri Krista Kiuru on antanut julkisuuteen useita lausuntoja käsillä olevasta muutoksesta. Kiuru kertoi Helsingin Sanomien 21.1.2014 julkaisemassa artikkelissa ohjelmoinnin tulevan peruskoulujen opetussuunnitelmaan. Ohjelmoinnista ei ole tulossa omaa oppiainettaan, vaan osa muiden oppiaineiden opetusta. (Lehtinen, 2014.) Yleisradiolle antamassaan lausunnossa opetusministeri Kiuru myöntää, että kansainväliset tutkimukset kertovat huolestuttavaa tietoa: Suomi on heikkoa tasoa kansainvälisessä vertailussa IT-opetuksessa. Yhteistyötä muiden maiden kanssa toimintamallien parantamiseksi on jo aloitettu muun muassa Viron kanssa, jossa IT-opetus aloitetaan tietyissä kouluissa jo ensimmäiseltä vuosiluokalta jatkuen lukion loppuun asti. (Malmberg, 2013.) Näihin esimerkkeihin, kuten muihinkin vastaaviin keskusteluihin, liittyy olennaisesti huoli suomalaisen teollisuuden rakenteellisesta muutoksesta, jossa perinteiset teollisuuden ammattitaidot eivät enää riitä työtehtävistä suoriutumiseen. Teknologian kehittyessä perinteiset teollisuuden toimialat menettävät merkitystään työllistäjänä, ja tilalle olisi kyettävä kehittämään tieto- ja viestintäteknologian alan työpaikkoja. Opetusministeri Kiuru myös toteaa, ettei Suomessa ole varaa jäädä tieto- ja viestintäteknologian osaamisessa yhtään enempää muita maita jälkeen. Esimerkiksi Singaporessa opetetaan ohjelmointia jo esiopetuksessa (Lehtinen, 2014.) Viron opetusministeri Jaak Aaviksoon toteaa Yleisradion artikkelissa: ”...nykymaailma vaatii käytännön taitoja, loogista ajattelua ja tietotekniikan hallintaa. Koodaaminen on erittäin tehokas keino opettaa niitä kaikkia” (Malmberg, 2013.)

Neittaanmäki, Lehto ja Kankaanranta (2014) vertailevat raportissaan ”Kohti laskennallisen ajattelun osaamista” ohjelmointia eri maiden opetussuunnitelmissa ja tekevät päätelmän,

että tieto- ja viestintätekniset taidot eivät ole riittävästi esillä koko koulutusjärjestelmässä. Kansalaisten digitaalinen lukutaito on puutteellista ja sekä EU:n että Suomen kansallisella tasolla on puute alan huippuosaajista. Raportin mukaan TVT-osaamistaidoissa ohjelmoinnin osalta tilanne tällä hetkellä on seuraava: Iso-Britanniassa ohjelmointitaitoja opetetaan ongelmanratkaisussa tarvittava määrä ja Yhdysvalloissa paneudutaan enemmän ohjelmoinnin luovaan käyttöön. Tanskassa puolestaan lähtökohta on ohjelmointikielten periaatteet ja ohjelmoinnin toteuttaminen. Esimerkiksi Iso-Britannian opetussuunnitelmassa ohjelmointi nähdään laskennallisten taitojen kehittämisen työkaluna, ohjelmointi nähdään osana ongelmanratkaisua. Computing programmes of study-ohjelman opiskelijoiden osaamistavoitteissa tietokoneohjelmoinnin hallinta ongelmanratkaisussa on analysointitaidon perusta. (Neittaanmäki et al., 2014, 24—33.) Edellä luetellut ovat muutamia esiin nostettuja esimerkkejä ohjelmoinnin opettamista koskevasta julkisesta keskustelusta.

1.2. Tutkimuksen tavoitteet, raja- ja rakenne

Tämän deskriptiivisen eli kuvailevan tutkielman tavoitteena on selvittää suomalaisten opettajien ohjelmointivalmiuksia sekä asenteita ohjelmoinnin opettamista kohtaan. Lähtökoh- tana tutkielman teolle toimivat teknologian hyväksymistä tutkivat mallit. Tutkielman tut- kimusongelmat ovat tutkielman tavoitteiden pohjalta seuraavat:

1. Millaiset valmiudet opettajilla on opettaa ohjelmointia peruskoulussa?
2. Millaiset asenteet opettajilla on ohjelmointia kohtaan?

Tutkielman aineisto on kerätty koodiaapinen.fi-sivuston järjestämän ohjelmoinnin MOOC- kurssin (Massive Open Online Course) osallistujille tarjotulla kyselyllä. Kyselyyn vastaa- minen ei ollut kurssin osallistujille pakollista, joten läheskään kaikki kurssille osallistujat eivät vastanneet kyselyyn. Tutkimuskysely järjestettiin siten, että vastaaminen oli mahdol- lista ennen kurssin alkua, jotta kyselystä saatava aineisto vastaisi osallistujien ennakkotie- toja ja -käsityksiä. Vastaajat edustavat jossain määrin ohjelmoinnista kiinnostuneita opetta- jia, koska he ovat ilmoittautuneet vapaaehtoiselle kurssille jo tässä vaiheessa, kun ohjel- moinnin opetuksen alkuun on vielä noin vuosi aikaa. Tutkimus, jonka tuloksiin tämän tut- kielman tuloksia soveltuvien osin vertaillaan, keräsi aineistonsa henkilöiltä, jotka ovat joko ”potentiaalisia ohjelmoinnin opettajia” (*potential teachers of programming*) tai ovat jo valmiiksi tietotekniikan opettajia (*teachers of ICT*) (Cowan, Oldham & FitzGibbon, 2014,

306.) Vertailututkimuksen vastaajien joukosta löytyy siis ainakin jälkimmäisen ryhmän osalta enemmän substanssiosaamista kuin tämän tutkielman vastaajilta keskimäärin.

Luvussa kaksi tarkastellaan ohjelmointia. Keskeisiä aiheita tässä luvussa ovat ohjelmointiin liittyvät käsitteet, näkökulmat ohjelmoinnin opetuksen hyödyistä sekä siitä, miten ohjelmoinnin oppiminen nähdään loogisen ajattelun kehittäjänä. Luvussa kolme tarkastellaan tutkielman teoreettisena viitekehyksenä toimivia keskeisiä teknologian hyväksymistä ennustavia malleja ja tutkielman viitekehyksenä toimivan tutkimuksen *The Programming Studio: Investigating teachers' readiness for teaching programming in the island of Ireland*, sekä tutkimuksen kontekstia ja eroja tähän tutkielmaan. Luvussa neljä esitellään tämän tutkielman toteutus: Tutkimusstrategia, aineistonkeruumenetelmä ja aineiston analyysi. Luvussa viisi esitellään tutkielmasta saadut tulokset ja luvussa kuusi avataan tutkimustuloksia syvällisemmin. Luvussa seitsemän esitellään keskeiset tutkimustulokset ja tutkielman yhteenveto.

2. Ohjelmointi

Opettajien ohjelmointivalmiuksien kartoittaminen ja asenteiden selvittäminen edellyttää, että ymmärretään mitä ohjelmointi on, sekä sen, miksi ohjelmointi halutaan osaksi perusopetusta.

Horowitzin (1983) mukaan tavallinen tietokoneen käyttäjä ei näe, mitä taustalla tapahtuu ohjelmia suorittaessaan, sillä tietokoneen toiminnot ovat hyvin pitkälti piilossa visuaalisen käyttöliittymän takana. Monelle voi tulla yllätyksenä, kuinka primitiivisiin komentoihin nykyaikainen tietokone perustuu. Tuotantolinjalta valmistuessaan ”tavallinen tietokone” ymmärtää vain aritmeettisia ja loogisia operaatioita, sekä joitain ohjauskomentoja. Näistä ominaisuuksista koostuu konekieli. Konekieli kuitenkin on kaukana ihmisen tavasta ajatella, joten ratkaisuksi on rakennettu niin kutsuttuja korkean tason ohjelmointikieliä. Näillä ohjelmointikielillä ohjelmoija voi kirjoittaa koneelle ohjelman, käyttäen ihmiselle loogista kieltä, joka sitten käännetään koneen ymmärtämälle konekielelle käyttäen käännösohjelmaa (Horowitz, 1983, 1). Tässä luvussa tarkastellaan ohjelmoinnin lisäksi ohjelmointiin liittyviä muita olennaisia käsitteitä, kuten ohjelmointikieltä ja ohjelmaa. Luvun loppuosa keskittyy tarkastelemaan ohjelmoinnin opettamiseen liittyviä näkökulmia, kuten ohjelmoinnin opetuksen hyötyjä.

2.1. Mitä on ohjelmointi?

Détiennen (2002) mukaan ohjelmointi on funktion teknisten tietojen muuntamista tekstimuotoiseksi ohjelmaksi, jonka kone osaa tulkita laskeakseen funktion (Détienne, 2002, 13.) Hyvönen et. al. (2012) mielestä ohjelmointi on yksinkertaisimmillaan toimintaohjeiden antamista ennalta määrätyn toimenpiteen suorittamista varten ja jokainen ihminen törmää arkielämässään jopa päivittäin ohjelmoinnin kaltaiseen toimintaan. Esimerkiksi kun ennalta vieraaseen paikkaan autolla ajavalle annetaan puhelimitse ajo-ohjeet: Ajo-ohjeet sisältävät sarjan komentoja ja ohjeita, jotka ohjaavat toimenpiteen suorittamista, lopputuloksena toivottavasti on onnistunut suoritus ja oikeaan paikkaan päätyminen. Alkeellista ohjelmointia ovat myös niin matkapuhelimen herätyskellon asettaminen herättämään aamulla, kuin mikroaaltouunin käyttö ruoan lämmittämiseksi oikeaan lämpötilaan. Ohjelmoinniksi ajateltavaa toimintaa on siis hyvin monenlaista. Ohjelmoinnissa työvälineiden valinta riippuu asetetun tehtävän ratkaisuun käytettävissä olevista välineistä. Matalan tason

työvälineillä tehdään töitä hyvin yksinkertaisilla käsitteillä ja ilmaisuilla kun taas pitkälle kehitetyt korkean tason työvälineet mahdollistavat työskentelyn käsitteillä ja ilmaisuilla, jotka parhaimmillaan muistuttavat luonnollisen kielen käyttämiä käsitteitä ja ilmaisuja (Hyvönen, Lappalainen & Lakanen, 2012, 2—3.) Seuraavissa luvuissa käsitellään seikka-peräisemmin ohjelmointiin liittyviä keskeisiä käsitteitä.

2.1.1. Tietokoneohjelma ja konekieli

Kun tarkastellaan ohjelmointia, aivan aluksi on syytä selvittää tärkeimpiä ohjelmointiin liittyviä käsitteitä, tärkeimpänä itse tietokoneohjelma. Détiennen (2002) mukaan syntaktisesti eli lauseopillisesti ohjelma on tekstiä, joka koostuu tarkoin määritellyistä kieliopillisista säännöistä. Semanttisesti eli merkitysoopillisesti ohjelma tarkoittaa laskelman kuvausta, tarkemmin määriteltynä, laskelmien kokonaisuuden kuvausta (Détienne, 2002, 13.) Paananen (2005) mukaan tietokoneohjelma on joukko toimintaohjeita eli käskyjä, jotka tietokoneen suoritin toteuttaa. Tietokoneohjelman sisältämästä toimintaohjeistuksesta voidaan käyttää nimeä algoritmi (Paananen, 2005, 130.) Ensimmäiset algoritmit ovat siis olleet konekielellä kirjoitettuja ohjelmia.

Harsun (2005) mukaan ensimmäisiä varsinaisia tietokoneita ohjelmoitiin suoraan konekielellä. Ohjelmointi konekielellä on sitä, että ongelmanratkaisualgoritmi muunnetaan binäärikoodiksi, eli ”nolliksi ja ykkösiksi”. Edelleen Harsun mukaan on selvää, että tällainen ohjelmointi on työlästä ja muutosten teko hankalaa. Esimerkiksi kaikki muistiosoitteet ovat viittauksia absoluuttisiin muistipaikkoihin ja koska uuden käskyn lisäys keskelle ohjelmaa aiheuttaisi ohjelman loppuosan siirtymisen koneen muistissa, täytyy myös kaikki loppuosaan kohdistuvat viittaukset päivittää, jotta ohjelma edelleen toimisi halutusti (Harsu, 2005, 22—23). Paananen (2005) määrittelee konekielen tietokoneen äidinkieleksi. Hänen mukaansa käytännössä konekieli on kokoelma numeerisia operaatiokoodeja, jotka esitetään tietokoneen muistissa kaksi- eli binaarijärjestelmässä. Konekielellä pystytään tekemään yksinkertaisia operaatioita, kuten hakemaan tietoa keskusmuistista, tallentamaan sitä keskusmuistiin sekä tekemään laskutoimituksia ja vertailua (Paananen, 2005, 162). Koska konekielinen ohjelmointi ei ole erityisen helppoa tai käyttäjäystävällistä, ohjelmointiin on kehitetty konekieltä parempia ohjelmointivälineitä eli ohjelmointikieliä.

2.1.2. Ohjelmointikieli

Jotta voimme ymmärtää, mitä ohjelmointi on, meidän täytyy tarkastella myös käsitettä ohjelmointikieli. Paanasen (2005) mukaan kaikki tietokoneohjelmat on kirjoitettu jollakin ohjelmointikielellä. Ohjelmointikielen avulla ihminen pystyy esittämään tietokoneelle tarkeitte tehtävät ihmisen ymmärtämässä muodossa (Paananen, 2005, 162). Majaranta (2002) puolestaan näkee ohjelmointikielen muistuttavan tyypillisesti enemmän matemaatiikkaa kuin luonnollista kieltä (Majaranta, 2002, 163.) Harsun (2005) mukaan ohjelmointikieli on ”merkintäjärjestelmä, joka kuvaa laskentaa sellaisessa muodossa, että se on sekä ihmisen että koneen luettavissa” (Harsu, 2005, 13.) Dershem ja Jippingin (1990) mukaan ohjelmointikielet eroavat luonnollisista kielistä muutamien selkein eroin: Kommunikaatio tapahtuu ihmisen ja tietokoneen välillä, vaikka ohjelmointikieltä voi myös joissain tapauksissa käyttää ihmisten väliseen kommunikaatioon. Toinen merkittävä ero on kommunikation sisällössä, joka ohjelmointikielessä tunnetaan ohjelmana. Ohjelmat ovat riittävän tarkasti kirjoitettuja ongelmien ratkaisuesityksiä, jotta ohjelman vastaanottaja voi tuottaa ratkaisun (Dershem & Jipping, 1990, 1). Ohjelmointikieli on siis kieli, joka on tarkoitettu ohjelmoijan käytettäväksi kuvaamaan prosessia, jolla tietokone voi ratkaista ongelman.

Mitchell (2004) määrittelee ohjelmointikielet tietokoneiden ohjelmoinnin ilmaisukeinoksi. Ideaali ohjelmointikieli mahdollistaa ohjelmien ytimekkään ja selkeän kirjoittamisen. Hyvällä ohjelmointikielellä kirjoitettu ohjelma on helposti luettavissa sekä ymmärrettävä rakenteeltaan myös suuremman skaalan ohjelmissa (Mitchell, 2004, 3.) Détienne (2002) näkee merkittävimäksi eroksi ohjelmointikielten ja luonnollisten kielten välillä sen, että ohjelmointikielet ovat lauseopillisesti erittäin yksiselitteisiä. Ohjelmassa ei voi olla pieniäkään virheitä, joita puhutussa tai kirjoitetussa kielessä saattaa joskus ilmetä, kuitenkin vastaanottajan ymmärtäessä kerrotun (Détienne, 2002, 19.) Hyvönen et. al. mukaan koneen ohjelmointi poikkeaa merkittävästi ihmiselle annettavista ohjeista. Kone ei osaa automaattisesti kysyä neuvoa ennalta arvaamattomassa tilanteessa. Se toimii täsmälleen niiden ohjeiden mukaan, jotka sille on annettu. Koneelle kirjoitetun ohjelman tulee olla hyvin tarkoin määritellyssä muodossa ja ohjelman on pyrittävä ottamaan huomioon kaikki mahdollisesti esille tulevat tilanteet (Hyvönen et al., 2012, 2).

Loudenin (2003) mukaan lähes kaikki käytössä olevat tietokoneet perustuvat John von Neumannin johtaman ryhmän 1940-luvun loppupuolella muodostamaan konemalliin, josta myöhemmin kehitettiin RAM-kone eli hajasaantikone. Loudenin mukaan mallille tyypillis-

tä on muisti, eli joukko samankaltaisia muistipaikkoja sekä prosessori, jolla on käytettävissään joukko rekistereitä. Prosessori lataa tietoa muistista rekistereihin, suorittaa loogisia ja aritmeettisia operaatioita rekistereillä, sekä tallettaa rekisterien arvot takaisin muistiin. Koneelle tehty ohjelma koostuu käskyjoukosta, jolla tehdään edellä mainittuja operaatioita, sekä kontrollikäskyistä, joilla määrätään seuraava suoritettava käsky (Louden, 2003, 2—36). Ohjelmoinnin alkutaipaleella, 1950-luvulla, tietokoneen fyysisen toteuttamisen vaikeus oli niin hallitsevaa, että ohjelmointikielet suunniteltiin pelkästään koneen toiminnan näkökulmasta. Myöhemmin ongelmat ovat kasautuneet tietokoneiden hyväksikäyttöön. Ohjelmistotuotannon pullonkaulana ei niinkään ole koneen kykenemättömyys suorittaa ihmisen antamia käskyjä, vaan ihmisen kykenemättömyys käskeä konetta. Käskynanto olisi huomattavasti helpompaa, jos ohjelmointikielet olisivat myös ihmisille sopivia eli tasoltaan mahdollisimman korkeita, eli käytössä olisi mahdollisimman korkean tason työkalut (Louden, 2003, 37—39). Edellä esitetyn perusteella voidaan sanoa, että ohjelmointikielelle on useita määritelmiä. Yhteistä näille määritelmille on muun muassa se, että ohjelmointikieli nähdään eroavan luonnollisista kielistä erityisesti yksiselitteisyytensä puolesta.

2.2. Näkökulmia ohjelmoinnin opetukseen

Tässä luvussa tarkastellaan sitä, miksi ohjelmointi halutaan tuoda osaksi *Perusopetuksen opetussuunnitelman perusteita* sekä ohjelmoinnin opetuksen merkitystä peruskoulun oppilaalle.

Binkley, Erstad, Herman, Raizen, Ripley, Miller-Ricci ja Rumble (2012) näkevät informaatio- ja tietotyön osuuden kasvavan yhä enemmän kehittyneissä maissa, samalla kun perustuotannon osuus vähenee. Koneet tekevät yhä suuremman osan manuaalisesta työstä ihmisten puolesta. Tulevaisuuden työelämätaidoissa korostuvat kommunikaatiotaidot, ongelmanratkaisutaidot ja innovatiivisuus. Oppilaat eivät voi luottaa siihen, että tämän päivän koulussa opituilla perustaidoilla pärjäisi tulevaisuuden työelämässä. Binkley et al. (2012) listaavat kymmenen keskeistä tulevaisuuden työelämätaitoa neljään kategoriaan, jotka ovat:

- Ajattelutavat (*Ways of Thinking*), joka sisältää muun muassa luovuuden, innovoinnin, kriittisen ajattelun ja ongelmanratkaisutaidot

- Työskentelytavat (*Ways of Working*), johon sisältyy kommunikaatiotaidot ja yhteistyötaidot
- Työkalut (*Tools for Working*), johon sisältyy informaatiolukutaito (*Information literacy*) ja tietotekniikan käyttötaidot
- Elämätaidot (*Living in the World*), johon sisältyy muun muassa kansalaisuustaidot, elämän- ja uranhallintataidot sekä kulttuurinen tietoisuus ja kompetenssi.

Binkley et al. kirjoittavat näistä edellä mainituista taidoista nimellä *21st century skills*, jotka joissain yhteyksissä on suomennettu tämän vuosisadan taidoiksi (Binkley et al., 2012, 17–19). Binkley et al. mallissa ei tosin huomioda ”konkreettista tekemällä oppimista”, jota esimerkiksi Lindh pitää oleellisena. (Lindh, 2014, 6)

Sajjanharin ja Faulknerin (2014) mukaan ohjelmoinnin oppiminen vaatii taitojen hierarkista oppimista. Ohjelmoinnissa tarvittavat taidot voidaan pääpiirteissään jakaa korkean ja matalan tason taitoihin. Korkean tason taidot ovat: Analyttiset taidot ongelman analysoimiseksi ja ratkaisukonseptin kehittämiseksi. Matalan tason ohjelmointitaidoilla puolestaan tarkoitetaan koodin kirjoittamiseen tarvittavia taitoja, joita tarvitaan, että ohjelmasta saadaan syntaktisesti virheetöntä (Sajjanhar & Faulkner, 2014, 53). Majarannan (2002) mukaan olennaista ei ole ohjelmoinnin oppiminen itsessään, vaan ohjelmointitehtävien tarjoama mahdollisuus oppia ratkaisemaan ongelmia sekä kehittää lapsen matemaattista ja loogista ajattelua (Majaranta, 2002, 173.) Harsu (2005) näkee, että ohjelmointikielten yleisten periaatteiden tuntemisella on vaikutusta muun muassa ajatusten esittämiskykyyn ja kielen toteutuksen ymmärtämiseen. Samoin kuin luonnollisten kielten oppimisessa, myös ohjelmointikielen oppimisessa yleiset tiedot kieliopillisista rakenteista edistävät asioiden omaksumista. Harsun mukaan ohjelmoija, joka osaa vain yhtä ohjelmointikieltä, valitsee mielellään tämän kielen ohjelmointitehtäviinsä eikä aina ole halukas oppimaan uusia kieliä. Ohjelmointikielten yleiset periaatteet tunteva ohjelmoija pystyy paljon objektiivisemmin tekemään vertailuja eri kielten välillä ja valitsemaan aina kuhunkin tarkoitukseen sopivimman kielen (Harsu, 2005, 14). Dershem ja Jipping listaavat tärkeimmiksi ohjelmoinnin opettelemisen syiksi muun muassa ongelmanratkaisukyvyn paranemisen, ohjelmointikielen tarkoituksenmukaisuuden ja hyötykäytön oppimisen, sekä uusien ohjelmointikielten helpomman oppimisen (Dershem & Jipping, 1990, 3–4.) Pekkala ja Sola näkevät, että tällä hetkellä vain pieni osa tietokoneiden käyttäjistä tuottaa sovelluksia, joita muut käyttävät. Tulevaisuudessa tuottajia, niin ohjelmistojen kuin muidenkin alojen tuottajia, tarvitaan enemmän. Erityisesti tarvitaan henkilöitä, jotka hallitsevat niin konekielen kuin korkeam-

man tasonkin ohjelmointikielet. Tulevaisuuden ohjelmointivälineitä on vaikea ennakoida, mutta lähes kaikki nykyiset ohjelmat perustuvat John von Neumannin mallin mukaiseen perinteiseen ohjelmakoodiin. Tämän koodin osaaminen on siis välttämätöntä, sillä tavanomainen ohjelmointi tuskin tulee katoamaan vielä pitkään aikaan (Pekkala, 2002, 183; Sola, 2002, 194.) Edellä mainituista lähteistä ei ollut havaittavissa kritiikkiä tai mainittu, voisiko ohjelmoinnin oppimisesta olla jossain yhteyksissä tai tapauksissa haittaa ohjelmoinnin opettelijalle.

2.2.1. Koodauksen opetus on ajattelun opettamista

Majorannan (2002) mukaan ohjelmointi kehittää oppilaan kognitiivisia kykyjä. Oppiessaan ohjelmoimaan oppilas oppii ajattelevaan loogisesti, täsmällisesti ja rakenteellisesti. Ohjelmointia opetellessaan oppilas oppii analysoimaan ja kehittämään omaa ajatteluaan. Samalla kehittyy myös ajattelu, sillä ohjelmoijan täytyy ajatella sitä, miten ajattelee. Lisäksi Majoranta näkee ohjelmoinnin kehittävän matemaattisia kykyjä ja käsitteiden ymmärtämistä (Majoranta, 2002, 163). Beard (1971) listaa Piaget'n ja Inhelderin mallin lapsen kognitiivisten taitojen kehitysvaiheiden kolme pääkautta seuraavasti:

1. Ensimmäinen vaihe on sensomotorinen vaihe, joka kestää syntymästä kielen muodostumiseen asti suunnilleen kahdeksantoista ensimmäisen elinkuukauden ajan.
2. Toinen vaihe, joka voidaan jakaa edelleen kahteen vaiheeseen.
 - a. Ensimmäinen näistä on esioperationaalinen vaihe, joka kestää noin kahdeksantoista kuukauden iästä noin kahdeksanteen ikävuoteen. Tämä vaihe voidaan edelleen jakaa kahteen vaiheeseen, jotka ovat: Noin neljän vuoden ikään kestävä esikäsitteellinen vaihe sekä intuitiivinen vaihe.
 - b. Toinen vaihe on konkreettien operaatioiden vaihe, joka kestää suurin piirtein kahdeksannesta ikävuodesta nuoruusikään.
3. Kolmas vaihe on muodollisten operaatioiden kausi, joka alkaa noin kahdentoista vuoden iässä ja on täysin kehittynyt suurin piirtein kolme vuotta myöhemmin.

Edelleen Beardin mukaan Piaget'n mallissa matematiikassa suuri osa ajattelusta on henkisiä toimintoja, joita kuvataan erilaisin merkein ja kuvioin. Lapsen loogisen ajattelun kehitys jatkuu, kun siirrytään konkreeteista operaatioista abstraktisempiin muodollisiin operaatioihin. Algebralliset operaatiot ovat vaikeita vielä konkreettien operaatioiden kaudella olevalle lapselle, jos niitä ei esitetäisi jollakin muulla tavalla, esimerkiksi visuaalisesti

(Beard, 1971, 28—32). Kuuselan (2000) mukaan Piaget’n mallin mukaisissa vaiheissa mainitut ikärajat, jolloin lapsen oletetaan kyseisen tason saavuttavan, perustuvat keskimääräisiin arvioihin ja siihen, että viimeinen, formaalisten operaatioiden taso ylipäättään saavutetaan. Myös kehityksen tasoluonne voi olla pulmallinen. Tapahtuuko lapsen kehitys tasoittain vai voiko lapsen kehitystä kuvata tasoittain? Kritiikistä huolimatta Piaget’n luokitus on Kuuselan mukaan kuitenkin yleisesti hyväksytty ja käytetty (Kuusela, 2000, 40—41).

Tämän tutkielman kannalta olennaiset Piaget’n teorian vaiheet ovat konkreettien operaatioiden tason lisäksi muodollisten (formaalisten) operaatioiden taso. Formaalisten operaatioiden tasoa voidaan pitää älykkään aikuisen ajatteluna, jossa ajattelulle luonteenomaisena piirteenä on kyky loogiseen päättelyyn, jolloin ajattelulle on ominaista hypoteesien laatiminen ja päättely, sekä päätelmien käyttäminen hypoteesin testaamiseksi. Eli ajattelun taitoja, joita ohjelmoinnissa tarvitaan (Donaldson, 1983, 162—163). Ohjelmointiprosessi on ongelmanratkaisuprosessi. Lavosen, Meisalon ja Latun (2001) mukaan tehokkaan ongelmanratkaisun prosessiin kuuluvat ongelman hahmottaminen, ongelmaan liittyvien faktojen tunnistaminen, tavoitteen asettaminen, vaihtoehtojen ideointi ja muodostus, vaihtoehtojen arviointi, lupaavimman vaihtoehdon valinta, testaaminen sekä ongelman uudelleen arviointi, joiden jälkeen prosessi aloitetaan uudestaan alusta (Lavonen et al., 2001, 21.) Edellä mainitun perusteella voidaan todeta, että ohjelmoinnin avulla on mahdollista harjoittaa niin ajattelun, kuin ongelmanratkaisutaitojen kehittämistä kuin myös kehittää matemaattisia kykyjä. Samalla kehitetään myös niin kutsuttuja tämän vuosisadan taitoja.

2.2.2. Ohjelmoinnin opetuksen lähestymistapojen tarkastelua

Flanneryn ja Bersin (2013) mukaan lapsen iälle sopivien ohjelmointityökalujen ja opetus suunnitelman avulla lapsi kykenee ratkaisemaan luovasti ongelmia ja kehittämään monitieteisiä taitoja sekä tietoja (Flannery & Bers, 2013, 81.) White ja Sivitanides ovat päätyneet johtopäätökseen, jossa todetaan ohjelmoinnin opettelu yksilön kehitysvaiheeseen nähden liian abstraktilla ohjelmointikielellä johtavan ylikuormittumiseen, ja vastaavasti liian ”helpolla” ohjelmointikielellä opettelu johtavan pitkästymiseen (White & Sivitanides, 2002, 59.)

Mannila, Peltomäki ja Salakoski (2006) mukaan ohjelmointi on yleensä nähty vaikeana aiheena ja tutkimukset ovat osoittaneet, että aloittelijat kohtaavat useita ongelmia opetellessaan ohjelmoimaan. Oppiessaan ohjelmoimaan oppijat joutuvat kohtaamaan niin yleisiä

ongelmanratkaisutaitoja vaativia haasteita kuin ohjelmointikielen syntaksiin ja semantiikkaan liittyviä ongelmia. Jos opiskelija aloittaa liian helpolla kielellä, voi vastaan tulla ylittämättömän muuri "oikeisiin" ohjelmointikieliin siirryttäessä. Ohjelmoinnin opetteluun päättävöitteet kuitenkin ovat Mannilan mukaan ohjelmoinnin perustaitojen hankkiminen ja algoritmisen ajattelun kehittyminen, sekä tuleviin opintoihin valmistautuminen (Mannila et al., 2006, 211—212). Semperen (2005) mukaan ratkaisua graafisesta ohjelmointiympäristöstä tekstipohjaiseen ohjelmointiin siirtymisestä tuleviin ongelmiin ei välttämättä edes tarvita, sillä lähes kaikki ihmiset hyötyvät laskennallisen ongelmanratkaisukyvyn kehitymisestä, mutta vain harvat tarvitsevat oikeita ohjelmointitaitoja. Semperen artikkelissa on otettu kantaa LogoBlocks-ohjelmointikieltä koskevaan kritiikkiin, jolla todettiin LogoBlocksin soveltumattomuus käyttöjärjestelmän ohjelmoimiseen, koska käytettävissä on alle 50 primitiivistä ohjelmointiblokkia. Vastaus kritiikkiin on yksinkertainen: Kaikki eivät aio ohjelmoida käyttöjärjestelmää (Sempere, 2005, 12).

Mittermeirin (2013) mukaan algoritmin kehittäminen vaatii sen ilmaisemista jollakin formaalilla kielellä. Tavallisesti tuo formaali kieli ymmärretään tekstinä tai osittain graafisena. Mittermeirin tutkimuksessa oli tutkittu esikouluryhmien selviytymistä yksinkertaisten algoritmien keksimisestä kun käytössä ei ole luku- tai kirjoitustaitoa. Esikouluikäisen tapauksessa ei yleensä oleteta kirjoitus- ja lukutaitoa sekä abstraktin ajattelun taitoja, joita ohjelmoinnissa normaalisti tarvitaan, joten aiemmin on oletettu tämän ikäisten olevan kykenemättömiä siihen. Mittermeir kuitenkin todistaa havainnoillaan, että abstraktin ajattelun rajoittuneisuus ei ole este, jos ongelmat asetellaan esikouluikäisen kokemuksia vastaavasti ja jos ratkaisut ovat riittävän lyhyitä sekä esitettävissä jossain muussa kuin kirjoitetussa muodossa, eli visuaalisesti tai puhuen (Mittermeir, 2013, 557).

Sajjanharin ja Faulknerin (2014) mukaan visuaalisten ohjelmointiympäristöjen tapa kätkeä ohjelmakoodi graafisiin elementteihin mahdollistaa helpomman lähestymisen ohjelmoinnin opetteluun, mikä on tärkeää oppilaiden motivoitumiselle. Graafiset ohjelmointiympäristöt sopivat heidän mukaansa myös korkean tason ohjelmointitaitojen opetteluun (Sajjanhar & Faulkner, 2014, 53.) Eid ja Millham (2012) ovat selvittäneet tutkimuksessaan, kummasta aloittamistavasta on enemmän hyötyä, visuaalisella ohjelmointikielellä aloittamisesta vai konsolipohjaisesta proseduraalisella ohjelmointikielellä aloittamisesta. Tutkimuksessa seurattiin eri vuosiluokkien ryhmien etenemistä useiden vuosien ajan, kun ryhmät etenivät monimutkaisempiin ohjelmointikieliin. Tutkimustuloksista oli tilastollisesti selkeästi pääteltävissä, että konsolipohjaisella ohjelmointikielellä aloittaneet opiskelijat pärjäsivät pa-

remmin ohjelmoinnin jatkokursseilla kuin visuaalisella kielellä aloittaneet. Tämän mukaan pelkästään graafisia ohjelmointiympäristöjä käyttäen kognitiivinen kehitys jää vähäisemmäksi ja abstraktien ohjelmarakenteiden käyttö voi tuottaa vaikeuksia. Korkeammalla kognitiivisen kehityksen tasolla olevat opiskelijat pystyvät omaksumaan proseduraalisen ohjelmoinnin kautta perustaidot, joiden avulla he tunnistavat visuaalisen ohjelmointikielen taustalla olevan koodin ja siten pystyvät hyödyntämään visuaalista kieltä paremmin (Eid & Millham, 2012, 176). Edellä mainitun perusteella voidaan todeta, että ohjelmointitehtävät on suunniteltava huolellisesti. Suunnittelussa on otettava huomioon oppilaiden osaamistaso ja ohjelmoinnille asetetut tavoitteet. Edellä mainitun perusteella voidaan todeta, että ohjelmoinnin aloittaminen on suotavaa konsolipohjaisella kielellä, jotta kuvailevien kielten piilossa pysyvät mekanismit tulevat tutuiksi ohjelmointia opettelevalle.

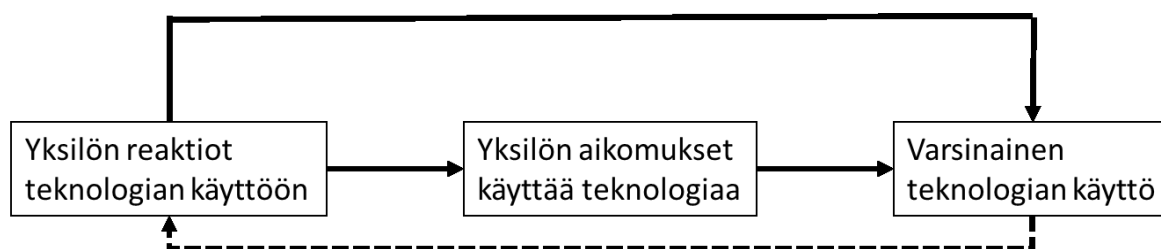
2.2.3. Yhteenveto

Yhdistävänä tekijänä edellä mainituille ohjelmointia käsitelleille tutkimuksille on se, että ohjelmoinnin oppiminen edellyttää peruskäsitteiden hallitsemista ennen kuin ohjelmointia voi hyödyntää ongelmanratkaisutyökaluna. Solan (2002) mukaan ohjelmoijan on joissain vaiheessa opittava ohjelmoinnin käsitteitä ja periaatteita ainakin implisiittisesti, eli siten, että ohjelmoija pystyy päättämään asiayhteyksiä. Näitä ohjelmoinnin periaatteita ovat muun muassa peräkkäisyys, valinta, toisto ja logiikka (Sola, 2002, 194.) Ohjelmointia oppimalla voidaan kehittää ongelmanratkaisutaitoja ja loogisen ajattelun kykyä. Samalla kehitetään myös muita tulevaisuuden työelämässä tarvittavia taitoja. Järvisen (2001) mukaan maailmanlaajuisessa kilpailussa menestyminen on kansallisen olemassaolon ehto. Taloudellinen menestys ja kilpailukyky teollisuuden eri aloilla säilyvät vain, jos tulevaisuuden työvoima on perillä teknologiaan liittyvien ongelmien ratkaisuprosesseista sekä pystyy innovatiiviseen ja monipuoliseen ajatteluun. Järvinen myös muistuttaa, että opettajat tekevät lasten kautta työtä tulevaisuutemme kanssa, joten vastuu on varsin suuri (Järvinen, 2001, 20—21). Kaiken kaikkiaan ohjelmointi nähdään tutkielman aineistossa ajattelua ja ongelmanratkaisukykyä kehittäväna toimintana. Tosin ohjelmointitehtävien suunnitteluun tulee panostaa.

3. Tutkimuksen viitekehys

Tutkielman lähtökohdat perustuvat kahteen keskeiseen näkökulmaan: 1. Opettajien valmiudet opettaa ohjelmointia, sekä 2.: Opettajien asenteet ohjelmoinnin opettamista kohtaan. Tutkielman tarkoituksena on tarkastella näitä näkökulmia sekä yksittäin että yhdistää näkökulmia toisiinsa.

Teoreettisen viitekehyksen tarkoituksena on avata teorioita, joiden lähtökohtana on selvittää miten ja miksi yksilöt omaksuvat uutta teknologiaa omakseen. Tutkielmassa keskitytään vain yksilön näkökulman avaamiseen. Molemmissa esitellyissä teorioissa keskeisin tekijä, joka vaikuttaa yksilön teknologian omaksumiseen ja käyttöön, on aikomus käyttää teknologiaa. Näissä teorioissa teknologian käyttö on kuvattu riippuvana tekijänä ja aikomus teknologian käyttöön nähdään käyttäytymistä ennustavana tekijänä. Kuva 1 esittää teknologian omaksumisen perusmallin ja lähtökohdat teorioiden avaamiseen (Venkatesh, Morris, Davis & Davis, 2003, 427).



Kuvio 1: Perusmalli teknologian omaksumisesta yksilötasolla (Venkatesh;Morris;Davis;&Davis, 2003, s.427)

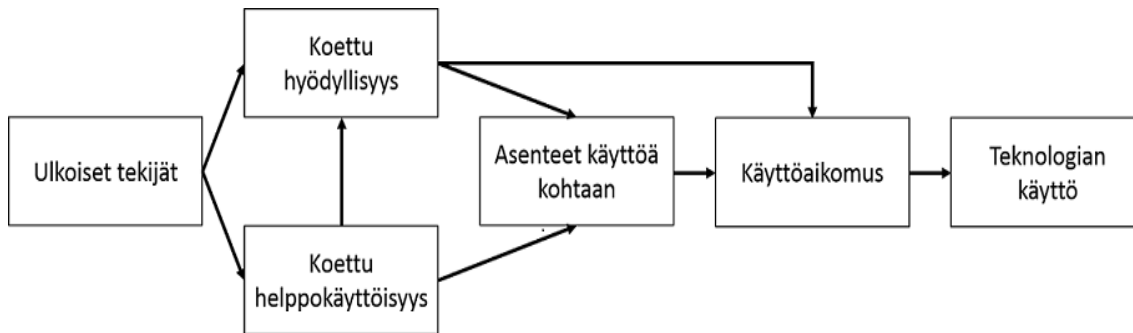
3.1. Teknologian hyväksymismalli

Tutkielman viitekehyksenä toimii TAM-teorian pohjalta Irlannissa tehty tutkimus *The Programming Studio: Investigating teachers' readiness for teaching programming in the island of Ireland* (Cowan et al., 2014.) TAM:n eli Teknologian hyväksymismallin (Technology Acceptance Model) avulla voidaan selittää ja ennustaa informaatioteknologian omaksumista yksilötasolla. Davis (1989) esitteli teorian ja sitä paranneltiin osittain Venkatesh:n ja Davis:n (2000) sekä Liu:n, Li:n ja Carlson:n (2010) toimesta. TAM perustuu Perustellun toiminnan teoriaan (Theory of Reasoned Action) eli TRA (Davis, Bagozzi & Warshaw, 1989, 985).

TRA on sosiaalipsykologinen malli, joka keskittyy selittämään tietoisien käyttäytymisen taustalla olevia tekijöitä. Käyttäytymisen ensisijainen vaikuttaja on aikomus käyttäytyä tietyllä tavalla. Aikomukseen vaikuttavat asenne ja subjektiivinen normi. Asenteen ja subjektiivisen normin muodostavat yksilön uskomukset, arvioinnit seurauksista sekä muiden ihmisten odotukset yksilön käyttäytymisestä (Davis et al., 1989, 983-984).

TAM:n mukaan yksilön *asenteisiin käyttöä kohtaan* (Attitude Toward Using), *käyttöaikomukseen* (Behavioral Intention) sekä varsinaiseen *teknologian käyttöön* (System Use) vaikuttaa useita muuttujia, mutta kaksi näistä vaikuttaa aikaisempien tutkimusten mukaan muita enemmän (Davis et al., 1989, 985; Davis F. D., 1989; Liu et al., 2010).

Ensimmäinen keskeinen muuttuja TAM:ssa on *Perceived usefulness* eli koettu hyödyllisyys. Koettu hyödyllisyys tarkoittaa astetta, jolla yksilö kokee, kuinka paljon teknologian käyttäminen auttaa häntä suoriutumaan työstään paremmin. Toisena keskeisenä muuttujana TAM:ssa on *Perceived ease of use* eli koettu helppokäyttöisyys. Koettu helppokäyttöisyys tarkoittaa yksilön uskomuksia siitä, kuinka paljon vaivaa hän joutuu näkemään teknologian käyttämiseen. Koettu hyödyllisyys vaikuttaa tutkimusten mukaan voimakkaammin kuin koettu helppokäyttöisyys. Nämä muuttujat vaikuttavat myös toisiinsa. Mikäli yksilö kokee tietyn teknologian helppokäyttöiseksi, pitää hän myös sitä itselleen hyödyllisenä. Mikäli ulkoiset tekijät, koettu hyödyllisyys ja helppokäyttöisyys koetaan neutraaleiksi, asenteen rooli tässä mallissa korostuu. Positiivinen asenne tiettyä teknologiaa kohtaan johtaa siihen, että yksilöt muodostavat itselleen aikomuksia ja aikomusten mukaisesti ottavat teknologian käyttöönsä. Kuvio 2 esittää keskeisten muuttujien suhteet asenteisiin, käyttöaikomukseen ja teknologian käyttöön (Davis, 1989, 320, 333—334; Davis et al., 1989, 985—986). Kuviossa 2 oleva laatikko, ”asenteet käyttöä kohtaan” tukee koetun hyödyllisyyden ja koetun helppokäyttöisyyden vaikutusta käyttöaikomukseen. Koettu hyödyllisyys vaikuttaa myös suoraan käyttöaikomukseen. Mikäli muut tekijät ovat neutraaleja, asenne käyttöä kohtaan määrittelee, kuinka paljon yksilö muodostaa aikomuksia käyttää teknologiaa. Muussa tapauksessa asenteet nähdään ulkoisina tekijöinä.



Kuvio 2: Teknologian hyväksymismalli (TAM) (Davis;Bagozzi;& Warshaw, 1989, 985)

Useiden tutkimusten perusteella on huomattu, että TAM selittää myös johdonmukaisesti suuren osan käyttöaikomusten ja käyttäytymisen vaihtelusta. Teknologian hyväksymismallista tuli ensimmäisten kymmenen vuoden aikana luotettava, voimakas ja tukeva malli ennustamaan teknologian hyväksymistä yksilötasolla. TAM:ia voidaan luotettavasti vertailla vastaaviin teorioihin, kuten TRA ja TPB (Theory of Planned Behavior) (Venkatesh & Davis, 2000, 186—187; Davis, 1989, 333). Haaparannan (2008) mukaan TAM-malli on yksinkertainen malli siitä, miten asenteet jotain tiettyä teknologiaa kohtaan muodostuvat. TAM-malli on kehitetty erityisesti sellaiseen työhön liittyvän teknologia-asenteen mittaamiseen, jossa työntekijällä on vapaus valita käyttääkö hän teknologiaa vai ei (Haaparanta, 2008). Opettajan työssä korostuu menetelmällinen vapaus, joten näin ollen TAM—malli sopii hyvin arvioimaan opettajan työhön liittyviä teknologia-asenteita.

3.2. Yhdistetty teoria teknologian hyväksymisestä ja käytöstä

Tämän tutkielman toinen taustalla oleva malli on UTAUT eli Yhdistetty teoria teknologian hyväksymisestä ja käytöstä (Unified theory of acceptance and use of technology) (Venkatesh et al., 2003, 446). Teknologian ja yksilön välistä suhdetta on tutkittu paljon ja teknologian hyväksymisestä on muodostettu useita erilaisia malleja. Jokaisella mallilla on omat lähestymistapansa, jotka pohjautuvat esimerkiksi psykologisiin tai ihmisten käyttäytymistä tutkiviin malleihin. Kaikki nämä mallit kuitenkin keskittyvät tutkimaan teknologian omaksumista. Tässä mallissa lähestytään yksilön teknologian omaksumista yksilön käyttöaikomusten kautta. UTAUT on kehitetty kahdeksan eri teorian, mm. TAM:n ja TRA:n pohjalta (Alwahaishi & Snásel, 2013, 63). Venkatesh:n (2003, 427—432) mukaan

UTAUT:n pohjalla olevat teoriat ovat:

- Theory of Reasoned Action (TRA)
- Technology Acceptance Model (TAM)
- Motivational Model (MM)
- Theory of Planned Behavior (TPB)
- Combined TAM and TPB (C-TAM-TPB)
- Model of PC Utilization (MPCU)
- Innovation Diffusion Theory (IDT)
- Social Cognitive Theory (SCT)

Alwanaishi et al. mukaan UTAUT ennustaa yksilön teknologian omaksumista ja käyttöönottoa (Alwahaishi & Snásel, 2013, 63; Venkatesh et al, 2003, 470.) UTAUT:ssa on onnistuneesti yhdistetty kaikkien kahdeksan pohjalla olevan teorian keskeisiä ydintekijöitä ja välillisesti vaikuttavia tekijöitä. Tähän malliin on tiivistetty yhteensä 32 ydintekijästä ja neljästä käyttäytymistä ja aikomusta määräävästä välillisestä tekijästä neljä keskeistä ydintekijää sekä neljä välillistä tekijää. Ydintekijällä tässä yhteydessä tarkoitetaan muuttujaa, joka vaikuttaa suoraan haluttuun ilmiöön eli tässä tapauksessa aikomukseen käyttää teknologiaa tai teknologian varsinaiseen käyttöön. Välillinen tekijä vaikuttaa haluttuun ilmiöön ydintekijöiden kautta (Venkatesh et al. 2003, 467.)

UTAUT mallin mukaiset ydintekijät ovat:

- Performance expectancy (Suoriutumisodotukset ohjelmoinnin opettamisesta)
- Effort expectancy (Vaivannäköodotukset)
- Social influence (Sosiaalinen vaikutus ohjelmoinnin opettamiseen)
- Facilitating conditions (Helpottavat olosuhteet)

Taustatekijöihin vaikuttavat välilliset tekijät ovat:

- Age (Ikä)
- Gender (Sukupuoli)
- Experience (Kokemus)
- Voluntariness of use (Käytön vapaaehtoisuus)

Ydintekijöistä suoriutumisodotukset, vaivannäköodotukset ja sosiaalinen vaikutus vaikuttavat suoraan teknologian käytön aikomukseen (Behavioral intention). Suoraan teknologian varsinaiseen käyttöön vaikuttaa käytön aikomuksen lisäksi myös helpottavat olosuhteet (Venkatesh et al., 2003, 467).

Tutkimusalueen sisällä on monia samaa asiaa kuvaavia teorioita, joista tutkijoiden pitää valita yksi, koska ei ole yhtä yhtenäistä teoriaa, jolla voisi mitata montaa ilmiötä samalla kertaa. Tällä valintatavalla muiden teorioiden hyvät puolet jäävät huomiotta ja ilmiön tutkimisesta tulee yksipuolista. UTAUT yhdistää onnistuneesti kahdeksan eri teorian keskeiset elementit ja vahvuudet yhdeksi malliksi. Teoria on pohjateorioiden määrään nähden yksinkertainen, mutta kuitenkin kattaa ilmiöstä useita eri näkökulmia. Esimerkiksi useat ilmiötä kuvaavat teorat eivät ota huomioon ikää taustamuuttujana. UTAUT:ssa se kuitenkin on mukana yhtenä välillisenä tekijänä, joka vaikuttaa ydintekijöihin ja sitä kautta teknologian käyttöön. Näin teorian avulla saadaan monipuolista dataa, jota voi soveltaa omien intressien mukaisesti (Venkatesh et al., 2003, 426, 467—471).

3.3. Muuttujien kuvaus

Muuttujat on rakennettu UTAUT-mallin mukaisesti ja suluissa oleva lyhenne ja termi liittyvät muuttujan TAM-malliin ja tämän tutkielman kontekstiin. Kaikki välilliset tekijät eivät vaikuta jokaiseen ydintekijään. Jokaisen ydintekijän kohdalla on mainittu siihen vaikuttavat välilliset tekijät.

Suoriutumisodotukset (PUT = Perceived Usefulness to Teachers)

Suoriutumisodotukset voidaan määritellä tasona, jolla yksilö uskoo, että teknologian käyttö voi parantaa hänen työsuoritustaan. Venkatesh et al. (2003) mukaan suoriutumisodotukset, joka pohjautuu kaikkiin kahdeksaan pohjateoriaan, on taustatekijöistä vahvin vaikuttaja

teknologian käytön aikomukseen. Kokemuksen ja käytön vapaaehtoisuuden ei nähdä vaikuttavan tähän ydintekijään, kun taas ikä ja sukupuoli vaikuttavat. Tämän tutkielman kontekstissa teknologian käyttö tarkoittaa ohjelmoinnin opettamista. (Venkatesh et al., 2003, 447)

Vaivannäköödotukset (PEU = Perceived Ease of Use)

Venkatesh et al. (2003) määrittelee vaivannäköödotukset asteena, kuinka paljon yksilö uskoo kokevansa vaivaa teknologian käytössä eli tämän tutkimuksen kontekstissa ohjelmoinnin opettamisessa. Välillisistä tekijöistä vapaaehtoisuus ei vaikuta vaivannäköödotuksiin ennakkokyselyssä, vaikka teknologian jo ollessa tuttua, vaikutus on merkittävä. Muut välilliset tekijät eli ikä, sukupuoli ja kokemus vaikuttavat yksilön vaivannäköödotuksiin (Venkatesh et al., 2003, 450).

Sosiaalinen vaikutus (SN)

Sosiaalinen vaikutus määritellään asteena, jolla yksilö havaitsee, kuinka paljon hänelle tärkeät ihmiset haluavat hänen käyttävän teknologiaa. Sosiaalinen vaikutus teknologian hyväksyntään on monimutkainen ja altis monille satunnaisille vaikutuksille. Sosiaalinen vaikutus voidaan jakaa kolmeen osaan: myöntyminen, sisäistäminen ja tunnistaminen. Kaksi jälkimmäistä vaikuttaa yksilön ajatusmaailmaan muuttumiseen tai sosiaalisen statuksen kehittymiseen. Tämän tutkielman kontekstissa SN nähdään positiivisena sosiaalisena tukena, eikä sosiaalisena paineena, jonka perusteella yksilön olisi pakko opettaa ohjelmointia, koska muut ihmiset vaativat sitä (Venkatesh et al., 2003, 451—453).

Helpottavat olosuhteet (FC)

Helpottavat olosuhteet määritellään asteena, jolla yksilö kokee, kuinka paljon organisatorinen ja tekninen infrastruktuuri tukee teknologian käyttöä. Helpottavat olosuhteet vaikuttavat suoraan teknologian käyttöön, koska ei aikomuksella ei nähdä kovin isoa roolia ympäröivän infrastruktuurin välittäjänä. Iällä ja kokemuksella nähdään isompi rooli olosuhteiden ja teknologian käytön suhteessa. Iäkkäämmät ja kokeneemmat yksilöt osaavat monipuolisemmin hakea organisaatiosta apua tai tukea esteiden ylittämiseksi ja jatkuvan käytön turvaamiseksi (Venkatesh et al., 2003, 453—454).

Henkilökohtaiset tekijät (A, AX ja SE)

UTAUT:n mukaan henkilökohtaiset tekijät (Personal factors) *asenne teknologiaa kohtaan* (Attitude toward using technology), *kyvykkyys* (Self-efficacy) ja *ahdistus* (Anxiety) eivät vaikuta suoraan teknologian käytön aikomukseen. Kyvykkyys ja ahdistus vaikuttavat käytön aikomukseen, mutta vain vaivannäköodotusten kautta, joten näitä ei voida pitää itsenäisinä ydintekijöinä. Asenne teknologiaa kohtaan kertoo yksilön kokonaisvaltaisesta ja tunneperäisestä reaktiosta teknologian käyttöä kohtaan. Tässä tutkielmassa kuitenkin tutkitaan myös opettajien asennetta ohjelmointia kohtaan, joten nämä pidetään epäsuorasti mukana. TAM-malli perustuu myös ajatukseen, että asenne teknologiaa kohtaan vaikuttaa suoraan käytön aikomukseen, mikäli muut muuttujat ovat neutraaleja (Venkatesh et al., 2003, 447, 455, 461).

3.4. Aiempi tutkimus aiheesta

Tutkielman pohjana toimii Irlannissa samasta aiheesta tehty tutkimus, joka pohjautuu teknologian hyväksymismalliin (katso 3.1.). Tämä tutkielma ei kuitenkaan ole täydellinen replikatutkimus Irlannissa tehdystä tutkimuksesta, sillä kyselylomakkeeseen on lisätty kyselyn mahdollistajan toiveesta osia UTAUT-teorian mukaisesti, jotta mahdolliset muut aineiston käyttäjät voivat hyödyntää aineistoa kattavasti. Tutkimuksien tuloksia vertaillaan soveltuvien osien tutkimustulokset-luvussa.

Kuten aiemmassa kappaleessa tuli ilmi, tutkielman alkuperäisenä pohjana toimi Irlannissa toteutettu tutkimus ”*The programming studio: investigating teachers’ readiness for teaching programming in the Island of Ireland*”. Tutkimuksessa on selvitetty kyselytutkimuksen avulla Irlannin ja Pohjois-Irlannin opettajien ohjelmoinnin opettamisen valmiuksia sekä asenteita ohjelmoinnin opettamista kohtaan. Tutkimuksen tuloksien perusteella tutkijat löysivät tarpeita opettajien koulutuksen laajentamiseen sekä opettajien ammatillisen identiteetin kehittämiseen ohjelmoinnin opettajiksi. Tutkimuksen taustalla on huoli opettajien valmiuksista toimia nopeasti digitalisoituvassa toimintaympäristössä. Tutkimus on osa projektia ”*The Programming Studio*”. *The Programming Studion* tarkoituksena on selvittää ja arvioida opettajien motivaatiota, halukkuutta, kompetenssia ja pystyvyyttä ohjelmointiin ja ohjelmoinnin opettamiseen koulussa. Tämän selvittämiseksi tehtiin tutkimus, joka dokumentoi kyseisiä asioita opettajilta, jotka ovat potentiaalisia ohjelmoinnin opettajia (*Potential teachers of programming*) Irlannista tai tietotekniikan opettajia (*teachers of ICT*)

Pohjois-Irlannista. Tutkimuksesta saatuja tietoja on tarkoitus hyödyntää pilottikokeilussa, jossa opettajat kokeilevat ”digitaalista lukutaitoa” opettelemalla yhden tai useamman ohjelmointikielen pelinomaisen oppimisen kautta. Tutkimuskysymyksekseen Cowan et al. asettivat kysymyksen: *”To what extent are prospective teachers of programming, in Northern Ireland and in the Republic of Ireland, able and willing to teach programming in schools?”* Tutkimuksen aineisto on soveltuvien osin analysoitu Technology Acceptance Modelin (TAM) mukaisesti (Cowan et al., 2014, 299—300).

3.5. Tutkimuksen konteksti: Ohjelmointi opetussuunnitelmassa

Ohjelmointi tulee osaksi perusopetusta uuden *Perusopetuksen opetussuunnitelman perusteiden* myötä. *Perusopetuksen opetussuunnitelman perusteissa 2014* koodaustaidot mainitaan osana laaja-alaisia osaamistavoitteita sekä ala- että yläkoulun puolella. Ohjelmointi näkyy opetussuunnitelman perusteissa 2014 luokka-asteittain seuraavasti:

Vuosiluokilla 1—2:

- Tieto- ja viestintäteknologinen osaaminen: *”Oppilaat saavat ja jakavat keskenään kokemuksia digitaalisen median parissa työskentelystä sekä ikäkaudelle sopivasta ohjelmoinnista”* (Opetushallitus, 2014a, 101.)
- Matematiikan tavoitteisiin liittyvät keskeiset sisältöalueet vuosiluokilla 1—2: *”Tutustuminen ohjelmoinnin alkeisiin alkaa laatimalla vaiheittaisia toimintaohjeita, joita myös testataan”* (Opetushallitus, 2014a, 129.)

Vuosiluokilla 3—6:

- Tieto- ja viestintäteknologinen osaaminen: *”Ohjelmointia kokeillessaan oppilaat saavat kokemuksia siitä, miten teknologian toiminta riippuu ihmisen tekemistä ratkaisuista”* (Opetushallitus, 2014a, 157.)
- Matematiikan opetuksen tavoitteet vuosiluokilla 3—6: *”Innostaa oppilasta laatimaan toimintaohjeita tietokoneohjelmina graafisessa ohjelmointiympäristössä”* (Opetushallitus, 2014a, 235.)
- Matematiikan tavoitteisiin liittyvät keskeiset sisältöalueet vuosiluokilla 3—6: *”Suunnitellaan ja toteutetaan ohjelmia graafisessa ohjelmointiympäristössä”* (Opetushallitus, 2014a, 235.)

- Matematiikan arviointikriteerit 6. vuosiluokan päätteeksi: Arvioinnin kohteet oppiaineessa on: ”*Ohjelmointi graafisessa toimintaympäristössä*” ja hyvä / arvosanan kahdeksan osaaminen on: ”*Oppilas osaa ohjelmoida toimivan ohjelman graafisessa ohjelmointiympäristössä*” (Opetushallitus, 2014a, 239.)
- Käsityön tavoitteisiin liittyvät keskeiset sisältöalueet vuosiluokilla 3—6: ”*Harjoitellaan ohjelmoimalla aikaan saatuja toimintoja, joista esimerkkinä robotiikka ja automaatio*” (Opetushallitus, 2014a, 271.)

Ohjelmointi kuuluu siis sekä matematiikan, että käsityön oppisisältöihin. Käsityön tavoitteissa ohjelmointia ei kuitenkaan ole nähtävissä, joten kyseessä on opetuksen eheyttämiseen ja monialaiseen oppikokonaisuuteen liittyvä näkökohta.

4. Tutkimuksen toteutus

Tutkimusstrategiana tutkielmassa on deskriptiivinen eli kuvaileva tutkimusote. Tutkielman aineisto on kerätty verkkokyselyllä koodiaapinen.fi-sivuston järjestämälle ohjelmointikursseille osallistuneilta opettajilta. Luvussa eritellään myös tutkimuksen kannalta tärkeiden summamuuttujien rakentamisprosessi. Tutkielman luotettavuuden ja eettisyyden arvioinnin kriteerit avataan luvuissa 4.7 ja 4.8.

4.1. Deskriptiivinen määrällinen tutkimus tutkimusstrategiana

Metsämuuronen (2005) ohjeistaa, että tutkimuksessa olisi hyvä valita joko kvantitatiivinen (määrällinen) tai kvalitatiivinen (laadullinen) tutkimusote päämetodologiaksi. Valinta tehdään tutkimuskohteen ja siitä saatavan tiedon pohjalta. Kvalitatiivinen tutkimusote on perusteltu, jos ollaan kiinnostuneita yksityiskohtaisesta tiedosta tai tutkitaan luonnollisia tilanteita, joita ei voida kontrolloida tai järjestää koetilanteeksi. Tässä tutkielmassa tutkitaan opettajien yleisluontoista asennetta ohjelmointia kohtaan sekä opettajien valmiuksia opettaa ohjelmointia. Opettajien mielipiteitä voidaan kysyä myös koetilanteessa, joten ei ole perusteltua käyttää kvalitatiivista tutkimusotetta vaan valitaan kvantitatiivinen lähestymistapa (Metsämuuronen, 2005, 202—203).

Tötön (2000) yksinkertaistavan näkemyksen mukaan kvalitatiivisella ja kvantitatiivisella tutkimusotteella saadaan vastauksia eri kysymyksiin ja tämän vuoksi tutkimuksen kysymysmuoto määrää käytettävän lähestymistavan. Kvalitatiivinen kuvaileva tutkimusote vastaa kysymykseen ”mitä?”, kun taas kvalitatiivinen selittävä tutkimusote vastaa kysymykseen ”miten?”. Kvantitatiivinen kuvaileva tutkimusote puolestaan vastaa kysymykseen ”miten paljon?” Tötön mukaan määrää voidaan ryhtyä mittaamaan vasta, kun on määritelty mitä mitataan, kuten esimerkiksi kun mitataan, ”miten yleinen jokin ominaisuus jossain ryhmässä on?” tai ”Onko ilmiö vähentynyt vai lisääntynyt?” Erityisesti jos kvantitatiivisen tutkimuksen varsinaisena tarkoituksena on ilmiön kuvaaminen, esitetään laadullisten ”miten?” – kysymysten sijasta määrällisiä ”miten paljon?” -kysymyksiä (Töttö, 2000, 74—76, 82).

Deskriptiiviset tutkimukset, kuten havaintotutkimukset ja kyselyt kohderyhmille, keskittyvät tarkkaan kuvailuun siitä, mitä tapahtuu (Lazar, Feng, Hochheiser, 2010, 21). Esimerkiksi tutkija voi tarkkailla yhtä luokkaa ja havaita, että kahdeksan kymmenestä oppilaasta,

jotka pelaavat määrättyä tietokonepeliä, osaavat kirjoittaa tietokoneen näppäimistöllä katsomatta näppäimistöä. Toisaalta vain kaksi kahdestatoista oppilaasta samalla luokalla, jotka eivät pelaa kyseistä peliä, osaavat kirjoittaa katsomatta näppäimistöä. Kyseessä on mielenkiintoinen havainto, mutta havainnon perusteella ei voida tehdä johtopäätöstä pelaamisen ja näppäimistöön katsomatta kirjoittamisen välillä, eikä tutkija voi myöskään selittää havaintojensa perusteella, miksi näin käy.

4.2. Verkkokysely aineistonkeruumenetelmänä

Tutkielman tutkimusinstrumenttina toimi verkkokyselylomake. Tekemällä tutkimuskysely verkossa, päästään helpommin käsiksi suurempaan määrään osallistujia. Toinen merkittävä hyvä puoli verkkokyselyssä on se, että vastaajien on helpompi pysyä anonyymeinä. Vastaajat voivat vastata silloin, kun heille sopii ja joissain tapauksissa verkkokyselyillä päästään käsiksi vastaajiin, joita ei muilla menetelmillä voitaisi tutkia, kuten esimerkiksi liikuntarajoitteisia. Verkkokysely ei välttämättä anna mahdollisuuksia tarkastella vastaajia erityisen tarkasti, mutta soveltuu hyvin kvantitatiiviseen lähestymistapaan ison otoskoon mahdollistajana (Lazar et al., 2010, 389—393).

Tutkielmaa varten kehitetyllä kyselylomakkeella kysytään myös asioita, joita tässä tutkielmassa ei käsitellä lainkaan. Syynä monipuolisen kyselylomakkeen kehittämiseen on se, että laajempaa aineistoa voisivat käyttää myös muut aiheesta kiinnostuneet tutkimuksen tekijät myöhemmin. Laajan kyselylomakkeen rakentaminen oli kyselyn järjestämisen mahdollistaneen koodiaapinen.fi-sivuston toimijoiden pyyntö. Kyselylomake on rakennettu UTAUT-mallin pohjalta sen monipuolisuuden ja soveltuvuuden johdosta. Kysymykset ovat kuitenkin sovellettavissa myös tutkielmassa käytettyyn TAM-malliin näiden kahden mallin samankaltaisuuden takia.

Kyselylomake on pilotoitu Oulun yliopistolla luokanopettajaopiskelijoilla syyskuussa 2015. Kyselylomake on suunniteltu siten, että sen avulla saadaan selvitettyä olennaiset tiedot ja näkemykset, jotka vaikuttavat halukkuuteen opettaa ohjelmointia. Kyselylomakkeen avulla selvitetään niin taustatiedot (sukupuoli, ikä, työkokemus, käytön vapaaehtoisuus), kuin opettajien pätevyys, tietotaso ja asenteet ohjelmointia kohtaan. Tämän tutkielman kannalta olennaisimmat asiat on koottu seuraavien osioiden alle.

Ohjelmointiosaaminen ja -kokemus

Tähän osioon kuuluvat kysymykset selvittävät vastaajien itse raportoimaa kompetenssia. Osiossa on listattuna ohjelmointiympäristöjä, joita vastaajat todennäköisesti tunnistavat tai ovat käyttäneet. Jokaista ohjelmointiympäristöä kohtaan oli useampia vastausvaihtoehtoja alkaen siitä, onko vastaaja koskaan kuullutkaan kyseisestä ohjelmointikielestä, aina siihen, jos vastaaja voisi opettaa kyseistä ohjelmointialustaa toisille opettajille. Toiseksi tässä osiossa tutkitaan opettajien valmiuksia ratkaista ohjelmoinnin opetukseen liittyviä ongelmia.

Näkemykset ohjelmoinnin asemasta perusopetuksessa

Tämän osion tarkoituksena on selvittää vastaajien näkemys siitä, kuinka tärkeänä vastaajat näkevät ohjelmoinnin sisällyttämisen osaksi *Perusopetuksen opetussuunnitelman perusteita*. Vastausvaihtoehdot kysymyksiin olivat ”Täysin eri mieltä”, ”Osittain eri mieltä”, ”En osaa sanoa”, ”Osittain samaa mieltä” ja ”Täysin samaa mieltä”.

Asenteet ohjelmointia kohtaan

Tämä osio perustuu sekä TAM:iin että UTAUT:iin, joista tähän osioon on poimittu kolmekymmentäkaksi Likert-asteikolle rakennettua kysymystä, jotka on käännetty suomen kielelle ja muokattu vastaamaan asennetta ohjelmoinnin opetusta kohtaan. Osion tarkoituksena on mitata opettajien yleistä asennetta ja ahdistusta ohjelmointia kohtaan sekä taustalla vaikuttavien tekijöiden vaikutusta teknologian käytön aikomukseen.

4.3. Aineiston keruu

Mittauksen kohteita voidaan kutsua havaintoyksiköiksi, tämän tutkielman tapauksessa havaintoyksikkö on suomalainen luokanopettaja. Kaikkien havaintoyksikköjen muodostamaa kokonaisuutta kutsutaan tutkimuksen perusjoukoksi (population). Metsämuurosen (2005) mukaan kyselyyn vastaajat voidaan valita tutkimukseen kahdella tavalla: satunnaisesti tai ei-satunnaisesti. Ei-satunnaisille otoksille on tyypillistä se, että koehenkilöt on valittu saatavuuden tai harkinnan mukaan, kun halutaan tutkia oleellisia henkilöitä (Metsämuuronen, 2005, 53).

Tässä tutkielmassa on käytetty otantatapana ei-satunnaista otantaa. Ohjelmointi tulee koskettamaan kaikkia suomalaisia peruskoulun opettajia. Kyselyn järjestäminen kaikille suomalaisille opettajille ei olisi ollut tarkoituksenmukaista vielä tässä vaiheessa, sillä opettaji-

en ohjelmointikoulutus on vasta alkutekijöissään ja osa kysymyksistä saattaisi olla ymmärrettävissä vain aiheesta kiinnostuneille. Koodiaapinen.fi-sivuston järjestämälle ohjelmointikurssille ilmoittautuneet opettajat edustavat henkilöitä, joita voi kuvailla potentiaalisiksi ohjelmoinnin opettajiksi, sillä he todennäköisesti tulevat opettamaan ohjelmointia syksyllä 2016 ja ovat vapaaehtoisesti ilmoittautuneet ohjelmointikoulutukseen. Verrokkitutkimuksessa ”*The programming studio: investigating teachers’ readiness for teaching programming in the Island of Ireland*” tutkijat Cowan et al. määrittivät irlantilaiset vastaajansa käsitteellä ”*potential teachers of programming*” (Cowan et al., 2014, 306), joten nämä ryhmät ovat jossain määrin vertailukelpoisia. Ohjelmointikurssille ilmoittautuneet peruskoulun opettajat saivat tutkimuskyselyn täytettäväkseen ennen kurssin aloittamista, joten tutkielmassa kerätty aineisto edustaa ennakkotietoja ja ennakkokäsityksiä. Tutkielmassa kerätty aineisto edustaa siis vain ohjelmointikurssille ilmoittautuneita opettajia.

Koodiaapinen.fi-sivuston järjestämälle kurssille ilmoittautuneet opettajat ovat ilmoittautumisensa perusteella ilmaisseet jonkinlaista kiinnostusta ohjelmoinnin opettamista kohtaan. Kurssilla oli 2572 ilmoittautunutta osallistujaa, joista noin 800 suoritti ensimmäisen viikon tehtävät ajallaan. Näistä noin kahdeksastasadasta kurssilaisesta 317 vastasi tutkimuskyselyyn. Verrokkitutkimuksen ”*The programming studio: investigating teachers’ readiness for teaching programming in the Island of Ireland*” tekijät saivat tutkimuskyselyynsä yhteensä 219 vastausta, joista käyttökelpoisia oli 161. Vastauksista 90 oli Pohjois-Irlannista, jossa kysely lähetettiin 200—250:lle tietotekniikan opettajalle ja loput 129 vastausta olivat Irlannista, tosin tutkijat eivät osanneet sanoa, kuinka monelle kutsu osallistua tutkimukseen oli lähetetty Irlannissa, sillä tutkijat eivät tiedäneet kuinka monta vastaanottajaa sähköpostituslistalla oli. Kyselylomake oli lähetetty CESI (Computers in Education Society of Ireland) -postituslistalle (Cowan et al., 2014, 306). Aineisto kerättiin sähköisellä kyselylomakkeella, jossa oli 40 kysymystä. Alustana toimii Webropol ja kysely on avoinna kaikille MOOC-kurssin osallistujille. Kyselyyn osallistuminen oli kurssin osallistujille vapaaehtoista. Aineisto on kerätty 8.10.2015 — 20.10.2015.

4.4. Aineiston käsittely

Alkuperäistä kerättyä tutkimusaineistoa täytyy käsitellä huolellisesti ja suurimmassa osassa tutkimuksista aineisto vaatii valmistelua ja siistimistä ennen kuin aineistoa voidaan analysoida. Aineiston järjestämiseen ja siistimiseen on useita syitä. Esimerkiksi jos vastaajat

ovat syöttäneet vastauksia manuaalisesti, voi vastauksissa olla virheitä ja vastaukset voivat olla epätäydellisessä muodossa. Toisaalta, jokin analyysimetodi tai ohjelmisto voi vaatia aineiston järjestelemistä ennalta määritellyyn muotoon. Jos virheitä tai epätäydellisyyksiä ei poisteta tai korjata, ne voivat sotkea koko aineiston (Lazar et al, 2010, 70). Ensimmäinen tehtävä aineiston keräämisen jälkeen on aineiston tarkastelu virheiden varalta. Tämä on erityisen tärkeää tehdä, kun vastaukset ovat ihmisten syöttämiä, sillä kaikki ihmiset tekevät virheitä. Kaikkia virheitä ei ole mahdollista tunnistaa, mutta tarkoituksena on minimoida virheiden vaikutus. Virheiden tunnistamiseen on useita eri tapoja, kuten eri kysymysten vertailu saman vastaajan osalta. Esimerkiksi, jos vastaaja on syöttänyt iäkseen 25, mutta kertonut työkokemuksekseen toisessa kysymyksessä 30 vuotta, on aihetta epäillä vastauksen oikeellisuutta. Jos epäilyksiä aiheuttavaa kohtaa ei pystytäkään tarkistamaan mitenkään, täytyy vastaus poistaa kokonaan aineistosta (Lazar et al., 2010, 71). Virheiden tarkastuksen lisäksi ennen testejä ja analyysijä täytyy negatiiviseksi asetettujen kysymysten vastaukset koodata vastaamaan muita samaa asiaa koskevia kysymyksiä, jotta vastaukset kuvaisivat haluttuja asioita samalla asteikolla (Metsämuuronen, 2005, 507—508).

Kerätyn aineiston siistimisen jälkeen on hyvä suorittaa muutamia tavallisimpia kuvailevia tilastollisia analyysieja, joilla selvitetään aineiston luonnetta. Esimerkiksi halutaan selvittää, millaiselle skaalalle vastaukset hajoavat (Lazar et al., 2010, 73; Metsämuuronen, 2005, 319, 323—325). Näitä tavallisimmin käytettyjä tarkastelutapoja, joita tässäkin tutkielmassa käytettiin, ovat keskiarvot, varianssit, keskihajonnat sekä -vaihtelu.

Tämän tutkielman kyselylomake on rakennettu siten, että osaan kysymyksistä on vastattava, jotta pääsee eteenpäin. Samoissa kysymyksissä voidaan valita vain yksi vaihtoehto kerrallaan. Näin minimoidaan tyhjien ja epätäydellisten vastausten määrä. Aineiston siistimisen jälkeen saatu data syötettiin SPSS-ohjelmaan, jolla suoritettiin aineiston analyysi.

4.5. Aineiston analyysi

Tämän tutkielman mielenkiinnon kohde on kerätyn aineiston kuvaileminen, joten ensisijaiset analyysimenetelmät ovat perustunnusluvut ja jakauman muotoon liittyvät testit. Kuvailevina analyysimenetelminä on käytetty frekvenssi- ja prosenttijakaumia, sekä sijaintia (keskiarvo) ja hajontaa (keskihajonta) kuvaavia tunnuslukuja. Verrokkitutkimuksessa *The Programming Studio: Investigating teachers' readiness for teaching programming in the island of Ireland* (Cowan et al., 2014) yhtenä analyysimenetelmänä on käytetty eksplora-

tiivista faktorianalyysii, jonka avulla Metsämuurosen (2004) mukaan etsitään muuttujien taustalla olevia latenteja muuttujia (Metsämuuronen, 2004, 17.)

Aineiston analyysimenetelmän valinnassa tulee huomioida analyysimenetelmän soveltuvuus tutkimuskysymykseen, sopivuus otantaan ja mittaustarkkuuteen, testin voimakkuus ja testin tehokkuus (Metsämuuronen, 2004, 16—17.) Aineiston muuttujien jakaumien normalisuus vaikuttaa analyysimenetelmien valintaan. Normaalisuutta tutkittiin käyttämällä usein suositettuun tapaan (Hair;Anderson;Tatham;& Black, 1998, 71—72) histogrammeja, yksiulotteisen jakauman tunnuslukuja sekä normalisuustestejä. Tässä tutkielmassa muuttujien jakaumat eivät täysin noudata normaalijakaumaa, mutta poikkeamat eivät ole merkittäviä. Tämän takia tutkielmassa on käytetty Pearsonin korrelaatiokerrointa. Jos ero olisi huomattava, olisi muuttujien korrelaatioita tarkasteltaessa käytettävä Spearmanin järjestyskorrelaatiokerrointa. Tutkielmassa käytetään pääkomponenttianalyysii faktorianalyysin sijaan, koska aineisto on vino, mutta otoskoko on yli 300 (Metsämuuronen, 2005, 544, 602). Faktorianalyysillä on kaksi perustarkoitusta. Ensinnäkin sen avulla voidaan tutkia muuttujia ja niiden välisiä yhteyksiä muuttujien taustalla olevia yhteisten tekijöiden tunnistamiseksi. Toiseksi sillä voidaan testata hypoteeseja muuttujien välisistä yhteyksistä (Kerlinger, 1993, 590).

Summamuuttujien riippuvuutta toisiinsa tutkitaan korrelaatiokertoimien kautta. Korrelaatiokerroin kertoo riippuvuuden voimakkuuden välillä -1 ja 1. Mitä lähempänä nollaa arvo on, sitä heikompi riippuvuus muuttujilla on. Arvon ollessa negatiivinen, toisen muuttujan arvon kasvaessa toinen pienenee. Positiivisella korrelaatiokertoimen arvolla toisen muuttujan arvo kasvaa toisen arvon kasvaessa. Mitä lähemmäksi korrelaatiokertoimen itseisarvo menee maksimiarvoa 1, sitä voimakkaampi riippuvuussuhde muuttujilla on (Metsämuuronen, 2005, 339, 345). Korrelaatiokertoimia voidaan kuvailla myös sanallisesti. Mikäli arvoksi saadaan 0,6—0,8, sanallinen kuvaus kuuluu ”Korkea riippuvuussuhde”. Arvoilla 0,4—0,6 suhdetta voidaan kuvailla ”melko korkeaksi” tai ”kohtuulliseksi”. Yli 0,8 korrelaatiokertoimet ovat harvinaisia ihmistieteissä, mutta siinä tapauksessa suhdetta voidaan kuvailla ”erittäin korkeaksi” (Metsämuuronen, 2005, 346). Korrelaation merkitsevyys (Significance) kertoo päätöksenteon riskitason, eli kuinka todennäköistä on, että yhteys ei olekaan todellinen eikä korrelaatio eroa perusjoukossa nollasta. Tässä tutkielmassa on käytetty yleisesti käytettyä riskitason maksimiarvoa 0,05. Käytännössä arvo tarkoittaa tuloksen tilastollista merkitsevyyttä ja kuinka mahdollista riippuvuus on saada sattumalta. Luotetta-

vassa tuloksessa sattuman mahdollisuus on alle 5 % (Metsämuuronen, 2005, 346—347, 415—416).

Tässä tutkielmassa keskeisimmät aineiston analyysimenetelmät ovat siis perustunnuslukujen ja jakaumien hajonnan kuvaileminen sekä rakennettujen summamuuttujien riippuvuuden kuvaileminen toisiinsa nähden.

4.6. Summamuuttujien rakentaminen

Tutkimuskyselyn kysymykset oli rakennettu siten, että kysymyksiä yhdistämällä voidaan rakentaa summamuuttujia kuvaamaan tutkimuksessa haluttuja latentteja piirteitä. Latentti piirre tarkoittaa yksilön ominaisuutta, jota ei voida suoraan mitata mittareilla. Ominaisuus kuitenkin saadaan mitattua, kun uskotaan yksilön toimivan ja vastaavan kyseisen latentin piirteen suuruuden mukaisesti. Tässä tutkielmassa asennetta ohjelmointia kohtaan voidaan pitää latenttina piirteenä. Asennetta ei voi suoraan mitata millään mittareilla, mutta se saadaan esille esittämällä kysymyksiä tilanteista, joissa yksilöön vaikuttaa yksi tietty latentti piirre eli tässä tapauksessa yksilön asenne ohjelmointia kohtaan (Metsämuuronen, 2005, 145—146).

Latenttien muuttujien tunnistamiseen käytettiin pääkomponenttianalyysiä. Metsämuuronen (2005, 598) mukaan pääkomponenttianalyysiä voidaan käyttää alustavana menetelmänä faktorianalyysissä faktoreiden määrän arvioimisen tukena ja faktorimallin rakentamisessa. Analyysin rotaatiomenetelmänä käytettiin Varimax-rotatiota. Faktorianalyysin avulla muuttujille saatiin muodostettua osatekijöiden vaihtelua selittävät peruskomponentit, jotka selittivät yli 50 % empiirisesti mitattujen muuttujien kokonaisvaihtelusta (Metsämuuronen, 2005, 103, 598, 725).

Summamuuttujien reliabiliteettia tarkasteltiin reliabiliteettianalyysin avulla ja summamuuttujista saatiin Cronbachin alpha-arvot, joiden täytyy ylittää arvo 0,6, jotta summamuuttujat olisivat luotettavia. Summamuuttujista poistettiin kysymykset, joiden ei katsottu mittaavan summamuuttujaan liittyvää asiaa. Poistaminen perustui Alpha-arvoon (Metsämuuronen, 2005, 515). Summamuuttujien Cronbachin alpha-arvot on kuvattu taulukossa 1. Taulukossa on myös kuvattu summamuuttujan muodostamiseen käytettyjen kysymysten määrä (*No. of items*), esimerkki kysymyksen asettelusta (*Specimen Item*) sekä summamuuttujan selittämän vaihtelun prosenttiosuus kokonaisvaihtelusta (*% of variance*). Tässä tutkielmassa

jokaiselle summamuuttujalle tehtiin faktorianalyysi erikseen eli *% of variance* kuvaa jokaista muuttujaa omana kokonaisuutena.

Taulukko 1: Summamuuttujien rakentaminen

Item	No. of items	Specimen Item(s)	Alpha	% of variance
PUT	4	”Ohjelmoinnin opettaminen on hyödyllistä työni kannalta”, Ohjelmoinnin opettaminen tekee opetuksestani tehokkaampaa”	0,714	54,3 %
PEU	4	”Minusta ohjelmoinnin opettaminen on helppoa”, ”Minusta tulee helposti taitava ohjelmoinnin opettaja”	0,779	60,3 %
BI	3	”Suunnitelmissani on opettaa ohjelmointia seuraavien (syötä numero) <n> kuukausien aikana”	0,895	83,6 %
SN	2	”Minulle tärkeiden ihmisten mielestä minun tulisi opettaa ohjelmointia”, Käyttäytymiseeni vaikuttavien ihmisten mielestä minun tulisi opettaa ohjelmointia”	0,798	83,2 %
A	4	”Ohjelmoinnin opettamista tarvitaan peruskoulussa”, ”pidän ohjelmoinnin opettamisesta”	0,769	59,4 %
AX	4	”Epäröin opettaa ohjelmointia, koska saatan opettaa sitä väärin”, ”Tunnen ahdistusta ohjelmoinnin opettamista kohtaan”	0,823	65,4 %

Kaikki summamuuttujiin käytetyt kysymykset ovat Liitteessä 1. Lomakkeelta saadun datan pohjalta on muodostettu kuusi tutkielman kannalta keskeistä summamuuttujaa:

- PUT: Perceived Usefulness to Teachers
 - Kysymykset 8—11: Suoriutumisodotukset ohjelmoinnin opettamisesta (UTAUT: Performance Expectancy)
- PEU: Perceived Ease of Use

- Kysymykset 14—17: Vaivannäköodotukset (UTAUT: Effort Expectancy)
- BI: Behavioral Intention
 - Kysymykset 38—40: Aikomuksesta opettaa ohjelmointia
- SN: Subjective Norm
 - Kysymykset 22—23: Sosiaalinen vaikutus ohjelmoinnin opettamiseen (UTAUT: Social Influence)
- A: Attitude
 - Kysymykset 18—21: Asenteet ohjelmoinnin käyttöä kohtaan
- Ax: Anxiety
 - Kysymykset 33—36: Ahdistus

Yllä olevassa listassa on kuvattu muuttujan lyhenne, teoriaan sidottu termin koko nimi, liitteestä löytyvät kysymykset, jotka koskevat tätä muuttujaa, sekä UTAUT-mallin vastaavat muuttujat, mikäli sellaisia on.

4.7. Tutkimuksen luotettavuuden arvioinnin kriteerit

Kvantitatiivisen tutkimuksen luotettavuutta on perinteisesti arvioitu reliabiliteetti- ja validiteetti-käsitteiden avulla. Reliabiliteetilla viitataan tutkimuksen toistettavuuteen, mittausinstrumentin tarkkuuteen tai täsmällisyyteen (Kerlinger, 1993, 405.) Käytännössä reliabiliteettia arvioidaan selvittämällä satunnaisen mittausvirheen määrää tai osuutta aineistossa: Mittaustulos on reliaabeli siinä tapauksessa, että satunnainen mittausvirhe on vähäinen (Ketokivi, 2009, 54.)

Validiteetilla viitataan tutkimuksen yleistettävyyteen ja siihen, missä määrin tutkimus käsittelee sitä asiaa mitä sen on haluttu käsittelevän ja sitä, että ollaanko tutkimassa sitä, mitä on tarkoitus tutkia (Kerlinger, 1993, 417.) Tämä tutkielma on luonteeltaan deskriptiivinen, eli tutkielma keskittyy tarkasteltavana olevan ilmiön havainnointiin. Luotettava tutkimus voidaan tunnistaa siitä, että sama tutkimus voidaan toistaa toisten tutkijoiden toimesta muualla saaden tuloksia, jotka ovat johdonmukaisesti vastaavia, luotettavia ja kestäviä. Suuri haaste ihmisiä tutkivissa tutkimuksissa verrattuna fysiikkaan, kemiaan tai biologiaan ovat ihmisten vastausten vaihtelut. Ihmisten vastaukset saattavat vaihdella tilanteesta riippuen, joten tämän kaltaiset tutkimukset ovat hieman vaikeampia toistaa toisten tutkijoiden toimesta (Lazar et al., 2010, 21, 57; Metsämuuronen, 2005, 109—110).

Metsämuurosen (2005) mukaan valmiin mittarin käyttäminen tutkimuksessa on erittäin suotavaa erityisesti sen vuoksi, että tulokset ovat vertailukelpoisia muiden samalla mittarilla saatujen tuloksien kanssa. Kunhan kyseessä on mittari, jonka reliabiliteetti ja validiteetti on jo testattu laajoilla ihmismäärillä (Metsämuuronen, 2005, 58). Tutkielman kyselylomakkeen suunnitteluun antoi suuntaviivat kyselyn mahdollistanut koodiaapinen.fi-sivusto. Tutkielman tutkimuskyselylomakkeen kehittämisessä tuli ottaa huomioon tämän tutkimuksen tarpeiden lisäksi koodiaapinen.fi-sivuston tarpeet sekä Aalto-yliopisto, joka hallinnoi kerättyä aineistoa. Tutkielman tekijät kehittivät kyselyn, jonka Aalto-yliopisto otti käyttöönsä. Kyselylomakkeen kysymykset muotoiltiin UTAUT- ja TAM-mallien vastaavista kysymyksistä, joita on käytetty aikaisemmin. Tutkielman luotettavuutta voidaan parhaiten arvioida vertailemalla tutkielman tuloksia soveltuvin osin aiemmin tehtyyn tutkimukseen *The Programming Studio: Investigating teachers' readiness for teaching programming in the island of Ireland* (Cowan et al., 2014).

Tutkielman validiteetin kannalta on tärkeää tarkastella, miten hyvin käytetty tutkimusmenetelmä soveltuu tutkimusongelman tarkasteluun. Tutkielman reliabiliteetin kannalta on tärkeää, miten ristiriidattomasti ja johdonmukaisesti aineisto on kerätty ja analysoitu sekä tutkielman tutkimustulokset tulkittu.

4.8. Tutkimuksen eettisyyden arvioinnin kriteerit

Ihmisten tutkimiseen liittyy usein haasteita. Tutkittavien täytyy olla sopivia tutkimukseen, heidän tulee olla tietoisia oikeuksistaan sekä heidän yksityisyyttään täytyy suojella. Tutkijoilla on suuri vastuu, jotta tutkimuksesta tulee eettisesti kestävä (Lazar et al., 2010, 376). National Commission for the Protection of Human Subjects of Biomedical and Behavioral Research (1979) on luonut ihmistutkimukselle kolme yleistä eettistä periaatetta, jotta tutkittavien turvallisuus, yksityisyys, oikeudet ja terveys säilyvät ennallaan. Ensimmäinen periaate on *ihmisten kunnioitus*. Tämä voidaan jakaa kahteen osaan. Ihmisiä täytyy kohdella autonomisina olentoina ja ihmisiä, joilla ei ole täyttä autonomiaa itsestään, täytyy suojella. Autonominen ihminen pystyy tekemään päätöksiä ja toimimaan omien tavoitteidensa pohjalta. On myös ihmisiä, joilta tämä kyky puuttuu johtuen fyysisestä tai henkisestä sairaudesta tai olosuhteista, jotka erittäin vahvasti rajoittavat yksilön itsenäisyyttä. Ihmisten kunnioitus ihmistutkimuksessa näkyy tutkittavien vapaaehtoisuudessa. Heitä täytyy myös informoida tutkimuksesta, jotta he voivat itse päättää osallistumisestaan tutkimukseen.

Toinen periaate on *hyväntekeväisyys*. Tutkijoiden tehtävänä ei pelkästään ole noudattaa tutkittavien toiveita, vaan varmistaa tutkittavien hyvinvointi. Periaate voidaan tiivistää kahden sääntöön: älä vahingoita tutkittavia sekä yritä maksimoida tutkittavien hyödyt ja minimoida haitat. Kolmantena periaatteena on *oikeus*. Termin mukaan tarkoituksena on jakaa hyödyt tasaisesti kaikille niille, joille se kuuluu, kuten myös tutkimuksesta tulevat haitat. Kukaan, ei tutkijat tai tutkittavat, saa saada tutkimuksesta enemmän hyötyä tai haittaa, kuin he ansaitsevat. Se, mitä jokainen tutkimukseen osallistuva ansaitsee, on erittäin hankala kysymys ja jää tutkimukseen osallistuvien pohdittavaksi. Näiden kolmen periaatteen pohjalta voidaan muodostaa tutkimukselle yleinen vaatimus: *tietoinen suostumus*. Tietoinen suostumus pitää sisällään tutkimukseen osallistuvien riittävät tiedot tutkimuksen syistä ja toteutuksesta, mahdolliset riskit ja sen, miten he saavat lisää tietoa tutkimuksesta. Näiden lisäksi tutkittaville pitää olla selvillä tutkimuksen vapaaehtoisuus. Tutkittavat voivat lopettaa tutkimukseen osallistumisen kesken ilman jälkiseurauksia (National Commission, 1979; Lazar et al., 2010, 381). Tutkimuksen toteuttajan täytyy huolehtia, että tutkittavat ovat selvillä seuraavista tutkimusta koskevista asioista, jotta tietoisuuden vaatimus toteutuu. Tutkittavilla täytyy olla saatavilla tutkimuksen otsikko ja tarkoitus, mistä tutkimuksessa on kyse, kesto, hyödyt ja haitat, vaihtoehdot tutkimukseen osallistumiseen, yksityisyyden suojaaminen, ylimääräiset kulut, osallistujan oikeudet, tutkimuksen tekijöiden yhteystiedot lisätietojen varalta ja lisätietojen hankinta (Michigan Tech, 2015; Lazar et al., 2010, 382—383).

Tässä tutkielmassa aineistonkeruu tapahtui verkkokyselyllä, joka eroaa muista aineistonkeruumenetelmistä, koska tutkijat eivät tapaa tai näe tutkittavia. Tässäkin tapauksessa tutkijoiden täytyy pitää kiinni yleisistä tutkimuksen tekemisen eettisyyden säännöistä. Joitain osia tietoisuuden suostumuksen osista voi olla hankala toteuttaa, koska tutkittavia ei tavata missään vaiheessa. Lazar et al. asettavat tutkijoille muutamia kysymyksiä, joita on oleellista tarkastella tutkimuksen aikana: Voidaanko tutkittavien yksityisyys suojata? Voivatko tutkittavat ottaa yhteyttä lisäkysymyksien varalta? Saavatko tutkittavat riittävästi tietoa tutkimuksesta ennen kyselyn aloittamista? (Lazar et al., 2010, 391—392.) Toisaalta verkkokyselyn avulla vastaajien tiedot voidaan pitää helposti anonymisinä.

5. Tutkimustulokset

Tässä luvussa esitellään tutkielmassa kerättyä aineistoa tämän tutkielman tutkimuskysymysten kannalta merkittävien osien. Ensimmäisessä alaluvussa esitellään vastaajista saadut tiedot: sukupuoli, ikä, työkokemus, luokka, käytön vapaaehtoisuus ja aiemmat ohjelmointipinnat. Toisessa alaluvussa esitellään ohjelmointiosaamista käsitelleet kysymykset. Kolmannessa saadut tulokset vastaajien näkemyksestä ohjelmoinnin asemasta perusopetuksessa ja viimeisessä alaluvussa käsitellään vastaajien asenteita ohjelmointia kohtaan.

5.1. Vastaajista saadut tiedot

Ohjelmointikurssille ilmoittautuneista opettajista tutkielman tutkimuskyselyyn vastasi 317, joista yksi ei vastannut sukupuolta koskevaan kysymykseen. Vastaajien sukupuolijakauma on esitetty taulukossa 2.

Taulukko 2: Vastaajien sukupuoli

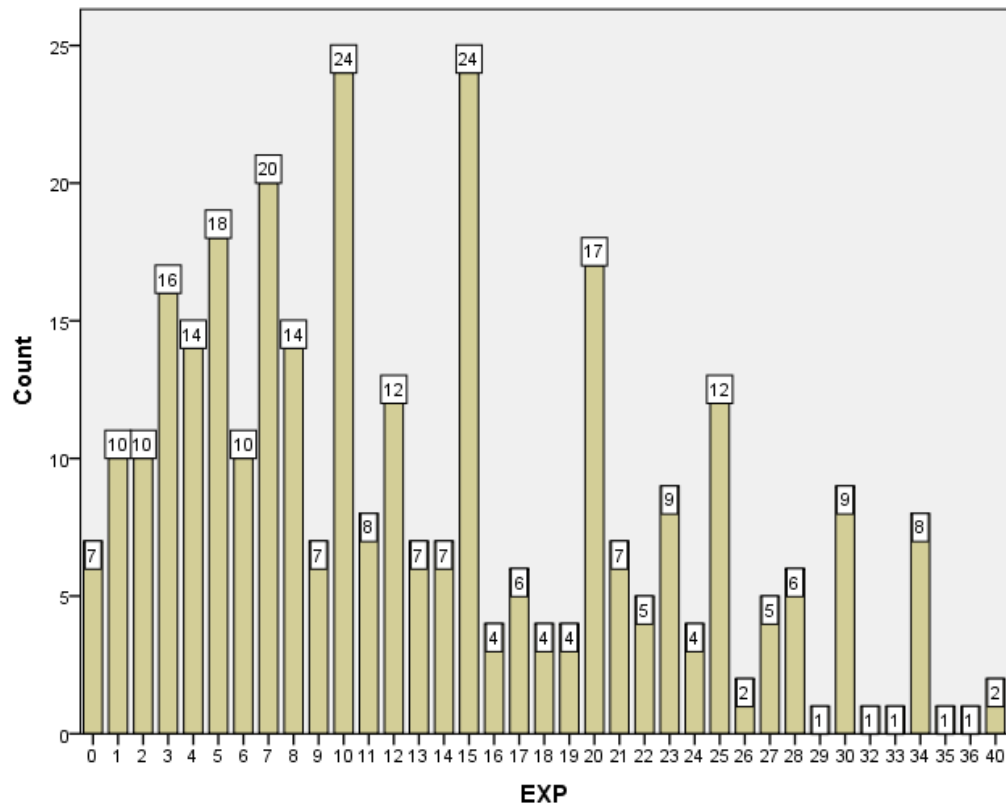
		Frequency	Valid Percent
Valid	Nainen	270	85,4
	Mies	46	14,6
	Total	316	100,0
Missing	System	1	
Total		317	

Vastaajat kertoivat ikänsä kyselylomakkeella valitsemalla pudotusvalikosta vastaajaa parhaiten kuvaavan vaihtoehdon. Vastausvaihtoehdot on numerokoodattu tilastollista käsitteilyä varten yhdestä (1= -15) kolmeentoista (13= 70+). Alle viisitoistavuotiaita, 20–24-vuotiaita sekä yli 70-vuotiaita vastaajia ei ollut lainkaan. Vastaajien jakautuminen ikäryhmiin on esitetty seuraavalla sivulla taulukossa 3.

Taulukko 3: Vastaajien ikäjakaumat

	Frequency	Valid Percent	Cumulative Percent
16-19	1	,3	,3
25-29	1	,3	,6
30-34	26	8,2	8,8
35-39	45	14,2	23,0
40-44	56	17,7	40,7
Valid 45-49	48	15,1	55,8
50-54	61	19,2	75,1
55-59	41	12,9	88,0
60-64	25	7,9	95,9
65-69	13	4,1	100,0
Total	317	100,0	

Työkokemukseen (EXP) vastaajat ilmoittivat keskimäärin 13,49 vuotta. Vastaajien työkokemus on esitettyä kuviossa 3. X-akselilla on vastaajien ilmoittamien työvuosien määrä ja Y-akselilla vastaajien määrä.

**Kuvio 3: Vastaajien työkokemus**

Vastaajat ilmoittivat luokka-asteen, jossa toimivat opettajana, valitsemalla pudotusvalikosta vastaajaa parhaiten kuvaavan vaihtoehdon. Vaihtoehdot olivat: ”1.—2. vuosiluokka”, ”3.—6. vuosiluokka”, ”7.—9. vuosiluokka”, ”Jokin muu” tai ”En opeta peruskoulussa”. Tutkimuskyselyä Webropol-alustalle siirtäessä tapahtuneen näppäilyvirheen vuoksi vastausvaihtoehdot on numerokoodattu 2= 1.—2. vuosiluokka, 3= 3.—6. vuosiluokka, 4= 7.—9. vuosiluokka, 5= Jokin muu, 6= En opeta peruskoulussa. Luokka-asteet, joilla vastaajat ilmoittivat työskentelevänsä, on esitetty taulukossa 4.

Taulukko 4: Vastaajien opettama luokka-aste

	Frequency	Valid Percent
1-2	68	21,5
3-6	110	34,7
7-9	98	30,9
Valid Jokin muu	21	6,6
En opeta peruskoulussa	20	6,3
Total	317	100,0

Käytön vapaaehtoisuutta kysyttiin lauseella ”Tulin kurssille, koska”, johon vastaaja valitsi pudotusvalikosta vastaajalle sopivan vaihtoehdon. Vastausvaihtoehdot numerokoodauksiin: 1= ”Työnantajani suositteli vahvasti”, 2= ”Työtoverini suositteli kurssia”, 3= ”Löysin itse ja kiinnostuin”, 4= ”Jokin muu syy”. Vastausten jakautuminen näille vaihtoehdoille on esitetty taulukossa 5.

Taulukko 5: Käytön vapaaehtoisuus

	Frequency	Valid Percent
Työnantajani suositteli vahvasti	10	3,2
Työtoverini suositteli vahvasti	62	19,6
Valid Löysin kurssin itse ja kiinnostuin	228	71,9
Jokin muu syy	17	5,4
Total	317	100,0

Vastaajien saamaa aiempaa ohjelmointikoulutusta selvitettiin kysymyksellä ”Onko sinulla loppututkintoa ohjelmoinnin opinnoista?” Vastausvaihtoehtoina tähän kysymykseen nume-

rokoodauksineen 1= ”Ei”, 2= ”Ei – mutta olen opiskellut itsenäisesti”, 3= ”Kyllä”. Vastaukset on esitetty taulukossa 6.

Taulukko 6: Vastaajien aiempi ohjelmointikoulutus

	Frequency	Valid Percent
Ei	267	84,2
Ei, mutta olen opiskellut itsenäisesti	33	10,4
Kyllä	17	5,4
Total	317	100,0

5.2. Ohjelmointiosaaminen ja -kokemus

Tähän osioon kuuluvat kysymykset selvittävät vastaajien itse raportoimaa kompetenssia. Osion ensimmäiseen kysymykseen oli listattuna ohjelmointiympäristöjä, joita vastaajat todennäköisesti tunnistavat tai ovat käyttäneet. Jokaista ohjelmointiympäristöä kohtaan oli useampia vastausvaihtoehtoja alkaen siitä, onko vastaaja koskaan kuullutkaan kyseisestä ohjelmointikielestä, jatkuen aina siihen, jos vastaaja kokisi voivansa opettaa kyseistä ohjelmointikieltä toisille opettajille. Vastaukset on esitetty taulukossa 7, joka on muotoiltu vastaamaan ulkoasultaan verrokkitutkimuksessa ”*The programming studio: investigating teachers’ readiness for teaching programming in the Island of Ireland*” julkaistua taulukkoa ohjelmointivalmiuksista. Koska vastaajat saattoivat halutessaan jättää kenttiä tyhjiksi, vastausten lukumäärä on ilmoitettu suluissa kunkin rivin ja vastausvaihtoehdon mukaan. Prosenttiosuus on laskettu kaikista vastaajista. Vastaukset on järjestetty sarakkeen ”*Voisin opettaa tätä kieltä*” mukaisesti suurimmasta pienimpään (Cowan et al., 2014, 306).

Taulukko 7: Vastaajien ohjelmointiosaaminen

<i>Ohjelmointiym- päristö</i>	<i>En ole kos- kaan kuullut tästä kielestä</i>	<i>Olen kuul- lut tästä kielestä</i>	<i>Olen käyttä- nyt tätä kieltä 3 viime vuo- den aikana</i>	<i>Harrastan ohjelmointia tällä kielellä</i>	<i>Voisin opettaa tätä kieltä</i>	<i>Voisin opettaa opettajille tätä kieltä</i>
<i>Scratch</i>	32,50 % (103)	37,20 % (118)	21,50 % (68)	0,30 % (1)	13,60 % (43)	8,50 % (27)
<i>Scratch JR.</i>	35,60 % (113)	41,60 % (132)	12,90 % (41)	0,60 % (2)	12,90 % (41)	7,30 % (23)
<i>Code.org</i>	47,60 % (151)	24 % (76)	16,70 % (53)	1,30 % (4)	10,10 % (32)	6,30 % (20)
<i>HTMLCSS</i>	12,60 % (40)	60,60 % (192)	18,90 % (60)	3,20 % (10)	7,30 % (23)	3,50 % (11)
<i>Lego NXT/RXT</i>	52,70 % (167)	25,90 % (82)	9,10 % (29)	0,60 % (2)	3,80 % (12)	2,50 % (8)
<i>Pascal</i>	36,60 % (116)	49,80 % (158)	1,30 % (4)	0,30 % (1)	3,20 % (10)	1,30 % (4)
<i>Racket</i>	51,40 % (163)	32,20 % (102)	4,40 % (14)	0,90 % (3)	2,80 % (9)	1,30 % (4)
<i>JavaScript</i>	10,10 % (32)	77,30 % (245)	8,20 % (26)	1,60 % (5)	2,80 % (9)	1,30 % (4)
<i>Java</i>	10,10 % (32)	74,40 % (236)	8,20 % (26)	1,60 % (5)	2,20 % (7)	0,30 % (1)
<i>BasicVisual Cool tms</i>	41,30 % (131)	45,70 % (145)	3,20 % (10)	0 % (0)	2,20 % (7)	0,90 % (3)
<i>Python</i>	46,40 % (147)	38,50 % (122)	5,40 % (17)	0,90 % (3)	1,60 % (5)	0,30 % (1)
<i>TurtleRoy</i>	68,80 % (218)	16,70 % (53)	3,20 % (10)	0 % (0)	1,30 % (4)	0,90 % (3)
<i>C/C++</i>	31,50 % (100)	54,90 % (174)	3,80 % (12)	0,30 % (1)	1,30 % (4)	0,30 % (1)
<i>Hopscotch</i>	68,50 % (217)	14,80 % (47)	5 % (16)	0 % (0)	0,90 % (3)	0,60 % (2)
<i>Logo</i>	66,60 % (211)	19,90 % (63)	0,90 % (3)	0 % (0)	0,90 % (3)	0,30 % (1)
<i>Processing</i>	80,10 % (254)	6,30 % (20)	0 % (0)	0,30 % (1)	0,60 % (2)	0,30 % (1)
<i>Cobol</i>	65 % (206)	19,60 % (62)	0,30 % (1)	0 % (0)	0,60 % (2)	0,30 % (1)
<i>PicaxeBasic</i>	78,90 % (250)	5 % (16)	1,30 % (4)	0,30 % (1)	0,60 % (2)	0,30 % (1)
<i>C#</i>	71,60 % (227)	13,90 % (44)	1,30 % (4)	0,60 % (2)	0,30 % (1)	0,30 % (1)
<i>AppInventor</i>	80,80 % (256)	4,70 % (15)	0,60 % (2)	0,30 % (1)	0 % (0)	0 % (0)
<i>Greenfoot</i>	81,40 % (258)	3,20 % (10)	0 % (0)	0 % (0)	0 % (0)	0 % (0)

Osion viimeisessä kysymyksessä kartoitettiin opettajien osaamista itsearviointikysymyksellä, jossa vastaajia pyydettiin valitsemaan sopiva (tai sopivat) vaihtoehdot jatkoksi lauseelle ”*Voin ratkaista ohjelmoinnin opetuksen ongelmia...*” Tällä tutkitaan opettajien kokemaa kyvykkyyttä opettaa ohjelmointia. Kysymys pohjautuu teorian Self-efficacy-kohtaan (SE). Vastausten jakautuminen on esitetty taulukossa 8. Prosenttiluku kullakin

rivillä ilmaisee osuutta kaikista vastaajista, vastaajilla oli mahdollista valita useampi vaihtoehto.

Taulukko 8: Opettajien ongelmanratkaisukyky

<i>Valitse kaikki sinulle sopivat väittämät jatkoksi lauseelle: "Voin ratkaista ohjelmoinnin opetuksen ongelmia..."</i>	<i>Frequency</i>	<i>Percent</i>
<i>"vaikka paikalla ei olisi ketään ohjeistamassa"</i>	142	44,8 %
<i>"jos voin soittaa apua ongelmatapauksissa"</i>	111	35,0 %
<i>"jos minulla on runsaasti aikaa perehtyä ongelmaan"</i>	230	72,6 %
<i>"jos minulla on käytettävissäni apua tai tukea"</i>	217	68,5 %
<i>ei mikään ylläolevista lauseista</i>	6	1,9 %

5.3. Näkemys ohjelmoinnin asemasta perusopetuksessa

Tämän osion tarkoituksena on selvittää vastaajien näkemys siitä, kuinka tärkeänä vastaajat näkevät ohjelmoinnin sisällyttämisen osaksi *Perusopetuksen opetussuunnitelman perusteita*. Vastausvaihtoehdot kysymyksiin olivat "Täysin eri mieltä", "Osittain eri mieltä", "En osaa sanoa", "Osittain samaa mieltä" ja "Täysin samaa mieltä".

Vastaajien näkemystä siitä, tarvitaanko ohjelmoinnin opettamista perusopetuksessa, selvitettiin väittämällä "Ohjelmoinnin opettamista tarvitaan peruskoulussa". Vastausten jakautuminen on esitetty taulukossa 9. Väittämä on liitteessä 1 numerolla 18. Vastausten keskiarvo on 3,95 ja keskihajonta 0,870.

Taulukko 9: Ohjelmoinnin opettamista tarvitaan peruskoulussa

	Frequency	Valid Percent	Cumulative Percent
Täysin eri mieltä	2	,6	,6
Osittain eri mieltä	18	5,7	6,3
En osaa sanoa	62	19,6	25,9
Osittain samaa mieltä	147	46,4	72,2
Täysin samaa mieltä	88	27,8	100,0
Total	317	100,0	

Väittämällä ”Työpaikallani suhtaudutaan positiivisesti ohjelmoinnin opetukseen” selvitetiin niin vastaajiin vaikuttavien sosiaalisten tekijöiden vaikutusta kuin myös näkemystä ohjelmoinnin asemasta omassa työyhteisössä. Vastausten jakautuminen väittämään on esitetty taulukossa 10. Kaikista vastaajista 169 valitsi vastausvaihtoehdon ”Osittain samaa mieltä” tai ”Täysin samaa mieltä”.

Taulukko 10: Työpaikallani suhtaudutaan positiivisesti ohjelmoinnin opettamiseen

	Frequency	Valid Percent	Cumulative Percent
Täysin eri mieltä	8	2,5	2,5
Osittain eri mieltä	45	14,2	16,7
En osaa sanoa	95	30,0	46,7
Osittain samaa mieltä	119	37,5	84,2
Täysin samaa mieltä	50	15,8	100,0
Total	317	100,0	

5.4. Asenteet ohjelmointia kohtaan

Tämä osion kysymykset perustuvat sekä TAM:iin, että UTAUT:iin. Tutkimuskyselyn tässä osiossa oli kolmekymmentäkaksi Likert-tyyppistä kysymystä, jotka oli muokattu ohjelmoinnin opetusta koskemaan ja käännetty suomeksi edellä mainittujen mallien mukaisista kysymyspatteristoista. Tarkoituksena oli kehittää asteikkoja, joilla voitaisiin mitata BI (Behavioral Intention), PUT (Perceived Usefulness to a Teacher), PEU (Perceived Ease of Use) ja SN (Subjective Norm). Edellä luetellut ovat TAM-malliin perustuvia muuttujia. Lisäksi summamuuttujiksi saatiin A (Attitude) ja Ax (Anxiety) mittaamaan yleistä asennetta ja ahdistusta ohjelmointia kohtaan.

Summamuuttujia mitattiin, lukuun ottamatta summamuuttujaa BI, Likert-asteikolla minimiarvona 1 ja maksimiarvona 5. Summamuuttuja BI mitattiin intervalliasteikolla minimiarvona 0 kuukautta ja maksimiarvona 99 kuukautta. Mittarilla pyrittiin kuvaamaan kuinka nopeasti opettajat aikovat ottaa ohjelmoinnin osaksi opetustaan (Metsämuuronen, 2005, s. 61.) Puuttuvia tietoja ei ole, vaan kaikki vastaajat ovat vastanneet kaikkiin kysymyksiin. Kaikille summamuuttujille saatiin keskiarvot, jotka kuvaavat opettajien keskimääräistä asennetta summamuuttujan kuvaamaan ilmiötä kohtaan. Summamuuttujien vaihteluväli, keskiarvot, keskihajonnat ja Cronbachin Alpha -arvot ovat taulukossa 11. Arvolla

5 PEU, PUT, SN ja A kuvaavat positiivista suhtautumista ohjelmoinnin opetukseen, kun taas AX ja BI kuvaavat samaa asiaa arvolla 1. Mitä pienemmän arvon jälkimmäisen ryhmän muuttujat saavat, sitä vähemmän tutkittavat esimerkiksi kokevat ahdistusta tai sitä nopeammin aloittavat ohjelmoinnin opettamisen. Ensimmäisen ryhmän kohdalla muuttujan suurimmilla arvoilla opettajat esimerkiksi kokevat ohjelmoinnin opettamisen positiivisena tai helppona oppia.

Taulukko 11: Summamuuttujien keskiarvot (N=317)

	Minimum	Maximum	Mean	Std. Deviation	Cronbac's α
PEU	1,00	5,00	3,23	,68501	0,779
PUT	1,00	5,00	3,12	,65045	0,714
A	1,00	5,00	3,74	,64213	0,769
SN	1,00	5,00	2,93	,90534	0,798
AX	1,00	5,00	2,19	,98819	0,823
BI	,00	99,00	7,60	10,42187	0,895
Valid N (listwise)					

Verrokkitutkimuksessa *The Programming Studio: Investigating teachers' readiness for teaching programming in the island of Ireland* (Cowan;Oldham;& FitzGibbon, 2014, s. 305) tutkittiin kyseisten summamuuttujien riippuvuutta toisiinsa ja teknologian varsinaiseen käyttöönottoon. Tämän tutkielman vastaavien summamuuttujien väliset korrelaatiot ovat kuvattu soveltuvien osin taulukossa 12.

Taulukko 12: Summamuuttujien korrelaatiot (N=317)

		PEU	PUT	BI	SN
PEU	Pearson Correlation	1			
	Sig. (2-tailed)				
PUT	Pearson Correlation	,376**	1		
	Sig. (2-tailed)	,000			
BI	Pearson Correlation	-,277**	-,264**	1	
	Sig. (2-tailed)	,000	,000		
SN	Pearson Correlation	,321**	,426**	-,171**	1
	Sig. (2-tailed)	,000	,000	,002	

** . Correlation is significant at the 0.01 level (2-tailed).

Kaikki korrelaatiot ovat tilastollisesti merkitseviä, eli niiden Significance-arvot ovat alle 0,05. Sattumalla on täten näissä korrelaatioissa alle 5 %:n vaikutus. Tilastollisesti merkitsevät tulokset on kuvattu taulukossa **-merkinnällä.

Tutkittaessa taustatekijöiden vaikutuksia summamuuttujiin, ei löydetty merkittäviä yhteyksiä. Sukupuolen vaikutukset näkyivät vain AX ja SN keskiarvoissa. Miesten keskiarvo AX muuttujalla on 1,8 kun naisilla vastaava on 2,3. SN kohdalla miesten keskiarvo 3,2 ja naisilla 2,9.

Erot taustatekijöiden vaikutuksista näkyivät myös vertaillessa aiemman koulutuksen vaikutusta keskeisiin muuttujiin. Vastaajat, joilla ei ole aikaisempaa koulutusta, kokivat hieman enemmän vaivaa (3,15), kuin vastaajat, joilla oli joko itsenäistä opiskelua (3,65) tai tutkintotodistus ohjelmoinnista (3,78). Aikaisempia opintoja suorittaneet kokivat myös ohjelmoinnin opettamisen positiivisempänä (4,05 & 4,22) kuin täysin ilman opintoja olevat vastaajat (3,67). Merkittävin ero tarkastelussa on ahdistuneisuuden ja koulutuksen tason välillä. Aikaisemmin opintoja suorittaneet kokivat ahdistuneisuutta keskiarvolla mitattuna alle 1,62, kun ilman opintoja olevien vastaajien kokema ahdistuneisuuden taso on 2,31. Kaikki keskiarvot opintotaustan mukaan jaoteltuna löytyvät taulukossa 13.

Taulukko 13: Ohjelmointikoulutuksen vaikutus summamuuttujiin

QUA	PEU	A	AX
Ei	3,15	3,67	2,31
Ei, mutta olen opiskellut itsenäisesti	3,65	4,05	1,51
Kyllä	3,78	4,22	1,62
Total	3,23	3,74	2,19

Tämän luvun tarkoituksena oli esitellä tutkimustulokset. Seuraavassa luvussa keskitytään tarkastelemaan tutkimustuloksia yksityiskohtaisemmin sekä vertaillaan tuloksia soveltuvien osin verrokkitutkimukseen.

6. Tutkimustulosten tarkastelu

Tässä luvussa tarkastellaan edellisessä luvussa esiteltyjä tuloksia. Tuloksia tarkastellaan soveltuvien osin verrokkitutkimukseen ”*The programming studio: investigating teachers’ readiness for teaching programming in the Island of Ireland*”, sekä Opetushallituksen raporttiin ”*Opettajat Suomessa 2013. Koulutuksen seurantaraportit 2014:8*”

6.1. Vastaajista saadut tiedot

Tutkielman tutkimuskyselyyn vastasi 317 henkilöä. Tutkimukseen vastanneista henkilöistä naisia oli 270 (85,4 prosenttia) ja miehiä 46 (14,6 prosenttia). Opetushallituksen seurantaraportin tilastoinnissa mukana olevien koulujen perusopetuksen rehtoreista ja opettajista naisia oli 73,6 prosenttia. Päätoimisia perusopetuksen kouluissa työskenteleviä opettajia oli 38179 (Opetushallitus, 2014b, 64, 67.) Naiset ovat siis suhteellisesti hieman yliedustettuina tämän tutkielman aineistossa. Verrokkitutkimuksessa ”*The programming studio: investigating teachers’ readiness for teaching programming in the Island of Ireland*” puolestaan vastaajia oli epätäydellisten vastausten siivoamisen jälkeen yhteensä 161, joista 100 oli miehiä (Cowan et al., 2014, 306.)

Vastaajien iän keskiarvo oli 8.11, eli 45—50 vuotta. Vastaajia oli kuitenkin kohtuullisen tasaisesti kaikista ikäryhmistä lukuun ottamatta alle 30-vuotiaita. Vastaajien joukossa oli yksi 16—19-vuotias vastaaja. Ikänsä puolesta tämä vastaaja ei vielä voi toimia opettajana, mutta tätä vastausta ei poistettu aineistosta sillä olettamuksella, että vastaaja on luokanopettajaopiskelija, joille MOOC-kurssia on markkinoitu. Opetushallituksen seurantaraportin mukaan perusopetuksen koulujen rehtoreista ja päätoimisista opettajista oli kevätlukukaudella 2013 alle 40-vuotiaita 31,0 prosenttia, 40—49-vuotiaita 32,8 prosenttia ja 50-vuotiaita tai vanhempia 36,2 prosenttia. (Opetushallitus, 2014b, 68.) Tutkielman vastaajien ikäjakaumassa on samankaltaisuutta opetushallituksen raportoimaan ikäjakaumaan. Opetushallituksen raportin mukainen suurin ikäluokka on suurin myös tämän tutkielman aineistossa.

Vastaajat edustivat kaiken pituisia työuria. Keskimääräisesti vastaajilla oli työkokemusta 13,49 vuotta mutta vastaajien joukossa oli runsaasti niin opettajia ilman työkokemusta kuin myös opettajia 40 työvuoden kokemuksella. 1.—2. vuosiluokan opettajia vastaajien joukossa oli 68, 3.—6. vuosiluokan opettajia 110, 7.—9. vuosiluokan 98 ja muita tai ei opetta-

jia yhteensä 41. Vastaajat edustivat siis varsin tasaisesti kaikkia perusasteen opettajia niin ylä- kuin alakouluistakin.

Käytön vapaaehtoisuutta selvittäneeseen kysymykseen suurin osa vastaajista ilmoitti löytäneensä kurssin itse ja kiinnostuneensa tai tulleen kurssille työtoverin suositeltua kurssia. Vain kymmenen vastaajaa ilmoitti työnantajan suositelleen kurssia vahvasti. Vastaajista 17 ilmoitti, että heillä on loppututkinto ohjelmoinnin opinnoista ja 33 ilmoitti opiskelleensa ohjelmointia itsenäisesti. Valtaosalla osallistujista ei siis ollut aiempaa kosketusta ohjelmointiin.

6.2. Ohjelmointiosaaminen ja -kokemus

Tutkielman tutkimuskyselyn vastaajat ilmoittivat tunnistavansa yleisimmin käytettyjä ohjelmointikieliä hyvin. Vastausvaihtoehdoista tosin puuttuivat yleisimpiin ohjelmointikieliin usein luettavat, mutta lähinnä raskaassa ammattikäytössä olevat Ruby, PHP, Objective-C ja SQL. Koska vastaajat saattoivat vastata useamman vaihtoehdon kunkin ohjelmointiympäristön kohdalta, jokaisen ohjelmointiympäristön kohdalta voi tarkastella vain yhtä vaihtoehtoa kerrallaan. Vastausvaihtoehdon ”Olen kuullut tästä kielestä” mukaisesti parhaiten tunnettuja olivat Javascript (245 vastausta), Java (236 vastausta), C/C++ (174 vastausta), Pascal (158 vastausta) ja HTML/CSS (192 vastausta). Yleisesti tunnettujen ohjelmointikielten tunnistaminen ei kuitenkaan tarkoittanut vastaajien keskuudessa sitä, että he olisivat ilmoittaneet harrastavansa ohjelmointia kyseisillä kielillä tai että he osaisivat opettaa kyseisiä kieliä. Vastausvaihtoehdon ”Voisin opettaa tätä kieltä” mukaan eniten osaamista löytyi graafisista ohjelmointiympäristöistä: Scratch (43 vastaajaa), Scratch Jr (41 vastaajaa) ja Code.org (32 vastaajaa). Heikoiten vastaajat tunnistivat lähinnä ohjelmointikielten opiskeluun tarkoitettuja ympäristöjä. Vastausvaihtoehdon ”En ole koskaan kuullut tästä kielestä” mukaan vähiten tunnettuja olivat Greenfoot (258 vastausta), AppInventor (256 vastausta), Processing (254 vastausta) ja PicaxeBasic (250 vastausta). Ohjelmoinnin harrastajia vastaajien keskuudesta ei juurikaan löytynyt, suurin määrä oli HTML/CSS-kielen kohdalla, jossa 10 vastaajaa (3,2 prosenttia vastaajista) oli valinnut vaihtoehdon ”Harrastan ohjelmointia tällä kielellä”. Scratchin kohdalla 68 vastaajaa (21,5 prosenttia vastaajista) ilmoitti käyttäneensä Scratchia viimeisen kolmen vuoden aikana, mutta vain yksi vastaaja ilmoitti harrastavansa ohjelmointia tällä kielellä.

Ongelmien ratkaisuun ohjelmoinnin opetuksen kontekstissa pystyisi vastaajista vastausten perusteella 72,6 %. Vain 1,9 % vastaajista ei kokenut pystyvänsä ratkaisemaan ongelmia millään tavalla. Tehokkaimmin opettajat kokevat pystyvänsä ratkaisemaan ohjelmoinnin opetuksen ongelmia, mikäli heillä on riittävästi aikaa perehtyä ongelmaan (72,6 %). Myös avun tai tuen saaminen ongelmien ratkaisuun (68, %) koettiin hyvänä vaihtoehtona ongelmien ratkaisuun. 44,8 % vastanneista opettajista kokivat myös pystyvänsä ratkaisemaan ohjelmoinnin opetuksen ongelmia ilman ulkopuolista apua tai tukea. 35,0 % vastaajista kokivat pystyvänsä ratkaisemaan ohjelmoinnin opetuksen ongelmia, mikäli apua tai tukea olisi saatavilla puhelimen välityksellä.

Verrokkitutkimuksen ”*The programming studio: investigating teachers’ readiness for teaching programming in the Island of Ireland*” vastaajista varsinkin Pohjoisirlantilaiset tietotekniikan opettajat tunnistivat yleisimpiä ohjelmointikieliä merkittävästi paremmin kuin tämän tutkielman suomalaiset vastaajat ja ilmoittivat useammin kykenevänsä opettamaan useimpia kieliä. Esimerkiksi HTML-kielen kohdalla Pohjois-Irlantilaisista vastaajista 40,3 prosenttia oli valinnut vastausvaihtoehdon ”*Could teach students*”. (Cowan et al., 2014, 306).

Tältä osin tutkimusten vertailu onkin järkevämpää tehdä vain verrokkitutkimuksen Irlannista raportoituihin vastauksiin opettajilta, jotka tutkijoiden mukaan ovat ”potentiaalisia ohjelmoinnin opettajia”. Jos verrokkitutkimuksesta jättää Pohjoisirlantilaiset vastaajat pois, kummassakin tutkimuksessa Scratch oli ohjelmointiympäristö, jolle löytyi eniten osaajia opettajiksi. Irlantilaisista vastaajista 19,2 prosenttia oli valinnut Scratchin kohdalla vastausvaihtoehdon ”*Could teach students*” kun tämän tutkielman suomalaisista vastaajista saman kielen kohdalla vastausvaihtoehdon ”*Voisin opettaa tätä kieltä*” valitsi 13,6 prosenttia vastaajista. Suomalaisista vastaajista tosin vain 37,2 prosenttia oli koskaan kuullut Scratchista kun irlantilaisten vastaajien keskuudessa vastaava luku oli 74,7 prosenttia. Cowan et al. (2014) tosin kirjoittavat, että osa heidän vastaajistaan oli todennäköisesti valinnut vain yhden vastausvaihtoehdon sen sijaan, että olisivat valinneet kaikki sopivat vastausvaihtoehdot. Tästä syystä luvut heidän tutkimuksessaan voivat olla alakanttiin muiden kuin ”*Heard of it*” vastausvaihtoehdon kohdalla. Toisaalta, suomalaisilla vastaajilla puolestaan oli ylimääräinen vastausvaihtoehto ”*En ole koskaan kuullutkaan tästä kielestä*”, jota verrokkitutkimuksessa ei ollut lainkaan. Toinen ohjelmointiympäristö, jolle kummassakin tutkimuksessa löytyi jonkin verran osaajia, oli HTML/CSS. Irlantilaisista vastaajista 9,1 prosenttia oli valinnut vastausvaihtoehdon ”*Could teach students*”. Tämän tutkielman

suomalaisista vastaajista 7,3 prosenttia valitsi vastaavan suomenkielisen vastausvaihtoehdon. Irlantilaisista vastaajista 84,8 prosenttia oli joskus kuullut HTML:stä ja suomalaisilla vastaava luku oli 60,6 prosenttia. Ainoa ympäristö, jossa osaamista löytyi suomalaisilta enemmän, oli Pascal. Suomalaisista vastaajista 3,2 prosenttia valitsi vastausvaihtoehdon ”Voisin opettaa tätä kieltä”, kun Irlantilaisten joukosta vastaavan vaihtoehdon valitsi 3,0 prosenttia vastaajista (Cowan et al., 2014, 306). Tekstissä mainittujen verrokkitutkimusta koskevien ongelmien vuoksi tarkempien vertailujen ja johtopäätösten tekeminen tältä osin ei ole tarkoituksenmukaista. Joka tapauksessa vastaajien raportoimat valmiudet opettaa tunnetuimpia ohjelmointikieliä eivät ole erityisen korkeita niin verrokkitutkimuksessa kuin tässä tutkielmassa.

6.3. Näkemys ohjelmoinnin asemasta perusopetuksessa

Tämän osion tarkoituksena on selvittää vastaajien näkemys siitä, kuinka tärkeänä vastaajat näkevät ohjelmoinnin sisällyttämisen osaksi *Perusopetuksen opetussuunnitelman perusteita*. Vastausvaihtoehdot väittämiin olivat ”Täysin eri mieltä”, ”Osittain eri mieltä”, ”En osaa sanoa”, ”Osittain samaa mieltä” ja ”Täysin samaa mieltä”.

Väitteessä, ”Ohjelmoinnin opetusta tarvitaan peruskoulussa”, täysin eri mieltä oli vain 2 vastaajaa (0,6 prosenttia vastaajista). Osittain eri mieltä oli 18 vastaajaa (5,7 prosenttia vastaajista) ja 62 vastaajaa (19,6 prosenttia vastaajista) vastasi ”En osaa sanoa”. Suurin osa vastaajista oli joko osittain samaa mieltä (147 vastaajaa — 46,4 prosenttia vastaajista) tai täysin samaa mieltä (88 vastaajaa — 27,8 prosenttia vastaajista). 235 vastaajaa 317:sta (74,1 prosenttia vastaajista) on siis osittain samaa mieltä tai täysin samaa mieltä väittämään ”Ohjelmoinnin opettamista tarvitaan peruskoulussa”.

Väittämällä ”Työpaikallani suhtaudutaan positiivisesti ohjelmoinnin opetukseen” selvitettiin ensisijaisesti vastaajiin vaikuttavien sosiaalisten tekijöiden vaikutusta. Kysymyksenasettelu antaa kuitenkin jossain määrin mahdollisuuden soveltaa väittämää myös vastaajan työyhteisöstään muodostaman näkemyksen tarkasteluun. Kaikista vastaajista 169 valitsi vastausvaihtoehdon ”Osittain samaa mieltä” tai ”Täysin samaa mieltä”. Kuitenkin lähes kolmannes vastaajista valitsi vastausvaihtoehdon ”En osaa sanoa”.

Verrokkitutkimuksessa ”*The programming studio: investigating teachers’ readiness for teaching programming in the Island of Ireland*” selvitettiin samankaltaisilla väittämillä ja

viisiportaisilla vastausvaihtoehdoilla vastaajien näkemystä ohjelmoinnin sisällyttämisestä opetussuunnitelmaan. Erona näissä tutkimuksissa oli se, että Cowan et al. selvittivät tutkimuksessaan niin vastaajiensa näkemyksen ohjelmoinnin sisällyttämisestä yleiseen opetussuunnitelmaan (*Curriculum*) kuin myös erikseen sitä, pitäisikö ohjelmointia opettaa seuraavilla asteilla: *Primary*, *Junior cycle minor*, *Junior cycle main*, *Senior cycle*. Cowan et al. tutkimustulosten mukaan Pohjois-Irlannin tietotekniikan opettajat suhtautuivat myönteisemmin ohjelmoinnin sisällyttämiseen niin kansalliseen opetussuunnitelmaan kuin myös kaikille edellä mainituille asteille. Myös irlantilaiset ”potentiaaliset ohjelmoinnin opettajat” suhtautuivat joko neutraalisti tai jokseenkin myönteisesti ohjelmoinnin sisällyttämiseen niin yleiseen opetussuunnitelmaan kuin kaikille asteille (Cowan et al., 2014, 307). Jotta vastaukset olisivat edes jossain määrin vertailukelpoisia tämän tutkielman vastaajien kanssa, täytyy verrokkitutkimuksesta jättää huomioimatta Pohjois-Irlannista saadut vastaukset, kuten aiemmissakin vertailuissa. Irlantilaisten ”potentiaalisten ohjelmoinnin opettajien” vastausten keskiarvo väittämiin, jotka koskivat ohjelmoinnin sisällyttämistä yleiseen opetussuunnitelmaan, oli noin 3,5, eli vastaajat suhtautuivat jokseenkin myönteisesti ohjelmoinnin sisällyttämiseen yleiseen opetussuunnitelmaan. Tämän tutkielman suomalaisten vastaajien vastausten keskiarvo väittämään ”Ohjelmoinnin opettamista tarvitaan peruskoulussa” oli 3,95, eli suomalaiset MOOC-kurssille osallistuneet opettajat suhtautuivat jopa huomattavasti myönteisemmin ohjelmoinnin sisällyttämiseen opetukseen kuin irlantilaiset ”potentiaaliset ohjelmoinnin opettajat”. Jos verrokkitutkimuksen Pohjois-Irlannista saadut vastaukset huomioidaan, voidaan sanoa, että suomalaisten vastausten keskiarvo tähän väittämään on suunnilleen sama kuin pohjoisirlantilaisten tietotekniikkaopettajien (Cowan et al., 2014, 307).

6.4. Asenteet ohjelmointia kohtaan

Tässä luvussa keskitytään tarkastelemaan luvussa 5.4 esitettyjä tuloksia tarkemmin. Tarkastelun alaisia muuttujia ovat BI (Behavioural Intention to teach programming), PUT (Perceived Usefulness to a teacher), PEU (Perceived Ease of Use) ja SN (Subjective Norm). Nämä ovat TAM-malliin perustuvia muuttujia. Lisäksi tarkastellaan summamuuttujista A (Attitude) ja Ax (Anxiety) saatuja tuloksia. Muiden muuttujien, lukuun ottamatta summamuuttujaa BI, keskiarvot on esitetty kuviossa 3. Kuten aikaisemmissakin tulosten tarkastelussa, on tarkoituksenmukaista verrata saatuja tuloksia vain verrokkitutkimuksessa Irlannista saatuihin tuloksiin. Verrokkitutkimuksessa ei myöskään avata muuttujien tarkko-

ja keskiarvoja, joten keskiarvot perustuvat tutkijoiden tekemiin arvioihin julkaistujen kuvioiden pohjalta (Cowan et al., 2014, 308).

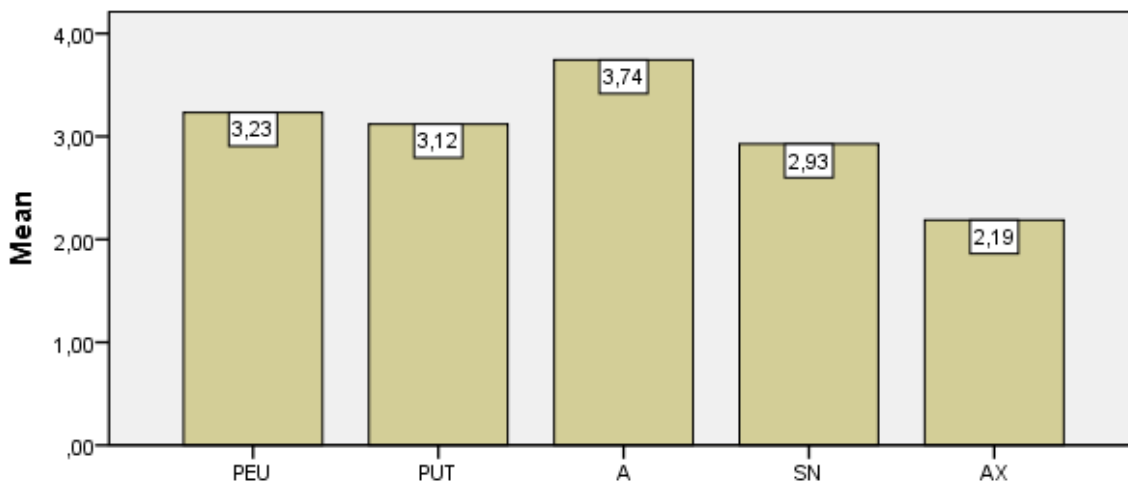
Perceived Usefulness to Teachers (PUT) muuttujan keskiarvo on 3,12. Kun kysymysten minimiarvo on 1 ja maksimiarvo 5, voidaan todeta, että opettajien kokemat suoriutumisodotukset sijoittuvat vain hieman neutraalin suhtautumisen positiiviselle puolelle. Opettajat eivät siis koe kovin vahvasti, että ohjelmoinnin opettamisesta olisi hyötyä heidän ammattitaitoonsa. Toisaalta opettajat eivät myöskään koe, että ohjelmoinnin opettamisesta olisi haittaakaan heidän työssään. Teoriaan verraten PUT:n pitäisi olla vahvin vaikuttaja teknologian käytön aikomukseen, mutta näiden tulosten pohjalta se ei sitä ole. Irlannissa saatu vastaava arvo on noin 3,75. Irlantilaiset opettajat siis kokevat ohjelmoinnin opettamisesta huomattavasti enemmän hyötyä kuin vastaavat suomalaiset.

Perceived Ease of Use (PEU) eli vaivannäköodotuksia koskevan muuttujan keskiarvo on 3,23. Opettajat kokevat, että he oppivat ohjelmoinnin opettamisen helposti. Arvo on kuitenkin lähempänä arvoa 3: ”en osaa sanoa”, kuin arvoa 4, jolla vastaajat olisivat jonkin verran samaa mieltä väittämien kanssa. Verrokkitutkimuksen arvo samalle muuttujalle on noin 2,8. Suomalaiset siis kokevat ohjelmoinnin opettamisen hieman helpompana oppia kuin verrokkitutkimukseen vastanneet irlantilaiset.

Subjective Norm (SN) eli Sosiaaliset vaikutukset -summamuuttujan keskiarvo on 2,93. Vastaajat siis kokevat, että heihin vaikuttavat sosiaaliset vaikuttajat olettavat heidän opettavan ohjelmointia. Vastausvaihto arvolla 3 on ”en osaa sanoa”, joten kyselyyn vastanneet opettajat eivät todennäköisesti osaa sanoa, kuinka paljon muut ihmiset vaikuttavat heidän käyttäytymiseensä ja täten ohjelmoinnin opettamiseen. Irlantilaisten tutkimuksen arvo tälle muuttujalle on noin 2,5. Tutkimukseen osallistuneet irlantilaiset kokevat siis saavansa suomalaisia kollegoitaan vähemmän sosiaalista tukea ohjelmoinnin opettamiseen.

Attitude (A) eli yleisestä suhtautumisesta ohjelmointia kohtaan summamuuttujan keskiarvo on 3,74. Opettajat suhtautuvat positiivisesti ohjelmointia kohtaan. Tämä tulos on odotettu, koska tutkielman otanta koski vapaaehtoisesti ohjelmoinnin opettamista käsittelevälle kurssille osallistuneita opettajia. On oletettavaa, että vapaaehtoisesti osallistuvilla asenne on hieman positiivisempi kuin niillä, jotka olisi pakotettu kurssille. Verrokkitutkimuksessa ei tutkittu vastaavaa muuttujaa, joten vertailua tämän tutkielman kanssa ei voi suorittaa.

Anxiety (AX) eli ahdistus ohjelmoinnin opettamista kohtaan —summamuuttujan keskiarvo on 2,19. Opettajat eivät osittain pidä ohjelmoinnin opettamista ahdistavana. Tähän muuttu- jaan liittyy samat lähtökohdat kuin A:n tuloksiin. Opettajat ovat osallistuneet vapaaehtoi- sesti kurssille, joten oli olettavaa, ettei ohjelmoinnin opetus ahdistanut tutkimukseen osallistu- neita kovin paljoa. Tätä muuttujaa ei myöskään tutkittu verrokkitutkimuksessa.

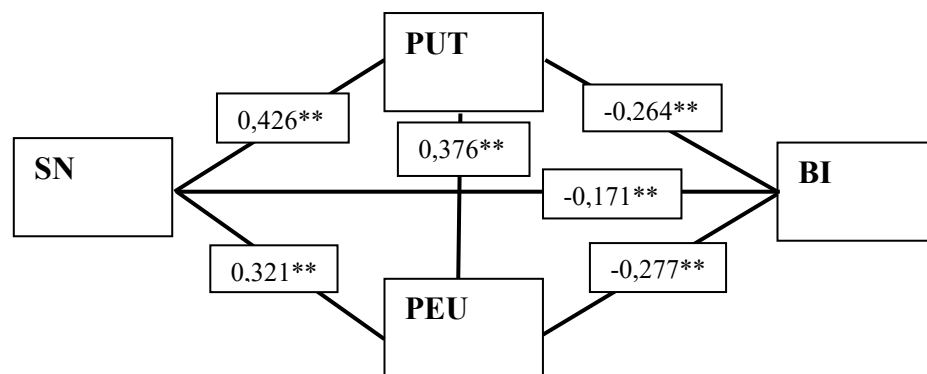


Kuvio 3: Keskeisten muuttujien keskiarvot

Behavioral Intention (BI) eli kuinka nopeasti opettajat aikovat ottaa ohjelmoinnin opetuk- seensa. Opettajat keskimäärin suunnittelevat ottavansa ohjelmoinnin opetukseensa 7,60 kuukauden kuluttua kyselyn täyttämistä. Kysymysten minimiarvo on 0 ja maksimiarvo 99 kuukautta. Verrokkitutkimuksessa tätä muuttujaa mitattiin Likert-asteikolla 1—5, joten näiden kahden vertaaminen suoraan ei ole kannattavaa. Korrelaatiotaulukosta kuviossa 4 on nähtävissä, että kaikki muuttujat ovat negatiivisessa yhteydessä Behavioral Intenti- on:iin. Kuitenkin verrokkitutkimuksessa ”*The programming studio: investigating teachers’ readiness for teaching programming in the Island of Ireland*” tutkijat saivat positiivisen korrelaatioarvon muiden muuttujien ja BI:n välille. Verrokkitutkimuksen ja tämän tutkiel- man kysymyksenasettelussa on eroa. Tässä tutkielmassa selvitettiin BI summamuuttujan kohdalla sitä, kuinka nopeasti opettajat aikovat ottaa ohjelmoinnin opetukseen käyttöönsä. Verrokkitutkimuksen kysymyksenasettelu koski sitä, kuinka helposti opettajat ottaisivat uutta teknologiaa käyttöön opetuksessaan (Cowan et al., 2014, 305).

Tässä kappaleessa avataan teoriaan (Katso 3.1) mukautettua mallia siitä, kuinka ydintekijät vaikuttavat toisiinsa ja teknologian käytön aikomukseen. Korrelaatiot on esitetty kuviossa 4. SN korreloi PUT:n ja PEU:n kanssa kohtalaisesti. Korrelaatiokerroin kertoo opettajien

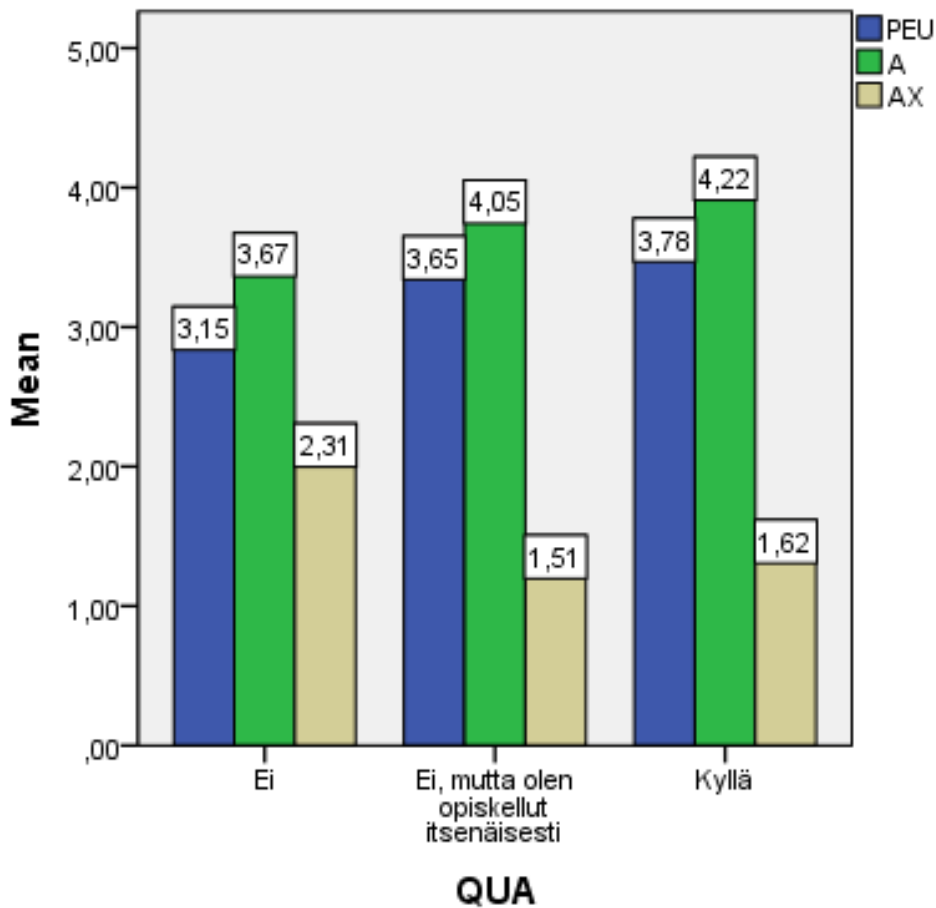
kokevan läheisten ihmisten hyväksynnän positiivisena sekä ohjelmoinnin käytön helppou-
teen ja suoriutumisodotuksiin. Kuten myös teorian oletuksena on, teknologian käytön
helppous vaikuttaa odotuksiin siitä, kuinka paljon teknologian koetaan auttavan käyttäjää
työssään. Opettajien kokema käytön helppous (PEU) vaikuttaa aikomukseen käyttää tek-
nologiaa (BI) suoriutumisodotuksia (PUT) enemmän. Käytön helppous kannustaa opettajia
ottamaan ohjelmoinnin opetukseensa aikaisemmin kuin sen koettu lisäarvo opetukseen.
Ero ei kuitenkaan ole merkittävä. Korrelaatioita tarkasteltaessa kokonaisuutena, voidaan
todeta kaikkien kolmen muuttujan (PUT, PEU ja SN) vaikuttavan aikomukseen ottaa tek-
nologia käyttöön negatiivisesti eli tämän tutkielman kontekstissa positiivisesti. Vastaajien
kokemat vaivannäköodotukset, suoriutumisodotukset ja sosiaalinen vaikutus siis nopeutta-
vat opettajien ohjelmoinnin opettamisen aloittamisen aikomusta.



Kuvio 4: Pearsonin korrelaatiot muuttujien välillä

Taustamuuttujien kautta tarkasteltujen keskeisten muuttujien keskiarvoissa esille nousi sukupuolen vaikutus AX ja SN -muuttujiin. Miesten vähäisen osuuden (14,5 prosenttia vastaajista) vuoksi erikseen miesten asenteista ohjelmoinnin opettamista kohtaan ei voida tehdä päätelmiä. Vähäisen otoskoon takia tätä ei ole tarkoituksenmukaista tarkastella enempää. Toinen taustamuuttuja, josta havaittiin eroavaisuuksia keskeisten muuttujien keskiarvoihin, on vastaajien aikaisemmat opinnot ohjelmoinnista. Ne vastaajat, joilla oli aiempaa opiskelukokemusta ohjelmoinnista, kokivat huomattavasti vähemmän ahdistusta ohjelmoinnin opettamisesta sekä asennoituivat positiivisemmin ohjelmointia kohtaan. Kuitenkin vain 15,8 prosenttia vastaajista ilmoitti suorittaneensa jonkinlaisia opintoja ohjelmoinnista, joten tässäkin tapauksessa otoskoko jää erittäin pieneksi ja lähempi tarkastelu ei

ole tarkoituksenmukaista. Ohjelmointikokemuksen vaikutus summamuuttujiin PEU (Perceived Ease of Use), A (Attitude) ja AX (Anxiety) on esitetty kuviossa 5.



Kuvio 5: Ohjelmointikokemuksen vaikutus summamuuttujiin

Tämän luvun tarkoituksena oli tarkastella ohjelmointiosaamisesta, näkemyksestä ohjelmoinnin asemasta sekä asenteista ohjelmointia kohtaan saatuja tuloksia, sekä tulosten vertailu verrokkitutkimuksen tuloksiin soveltuvien osien. Seuraavassa luvussa pohditaan tarkemmin keskeisiä tutkimustuloksia.

7. Yhteenveto ja pohdinta

Johdannossa asetettiin tutkielman perusteluiksi kysymykset: ”Millaiset valmiudet opettajilla on opettaa ohjelmointia peruskoulussa?” ja ”Millaiset asenteet opettajilla on ohjelmointia kohtaan?”

Tutkielman kirjallisuuden perusteella ohjelmoinnin syvällinen ymmärtäminen ei ole olennaista perusasteen oppilaalle tai opettajalle. Ohjelmoinnin perusteiden oppimiselle ja ymmärtämiselle kuitenkin löytyy useita syitä. Usein korostetaan esimerkiksi lukemisen tärkeyttä lapselle sillä, mitä valmiuksia se antaa elämää varten. Niin nykyään kuin tulevaisuudessaakin teknologian ja ohjelmoinnin ymmärtäminen yleisesti on samalla tavalla tärkeää, jotta pärjää työmarkkinoilla. Tämän vuoksi ohjelmoinnin opetus on varsin perusteltua. Opetussuunnitelmauudistuksen myötä olennaista ei ole tehdä kaikista lapsista ohjelmoinnin ammattilaisia, vaan tarjota mahdollisuus oppia ratkaisemaan ongelmia sekä kehittää matemaattista ja loogista ajattelua.

Keskeisiä asioita ohjelmoinnin perusasteen oppisisällöissä eivät ole ohjelmoinnin eri suuntauksat tai luokittelut, vaan käytännölliset harjoitukset ja ohjelmoinnin sitominen oppilaan arkielämään sekä ohjelmoinnin peruslogiikan ymmärtäminen. Tällaista perustietoa on kirjoittajien mielestä muun muassa se, että oppilas ymmärtää ihmisajattelun ja -kielen olevan käsitteellistä, mutta tietokone ei ymmärrä näitä ihmisten käsitteitä. Kone tarvitsee yksiselitteiset määritelmät voidakseen suorittaa haluttuja toimenpiteitä. Mikään ohjelmointikieli ei voi sisältää yksiselitteisiä määrittelyjä kaikille maailman käsitteille, eikä kone osaa automaattisesti kysyä neuvoa uuden, ennalta arvaamattoman tilanteen edessä. Kone tekee vain ja ainoastaan sen, mitä ihminen on koneen ohjelmaan kirjoittanut. Ohjelmointia harrastavia oppilaita löytyy kuitenkin mitä todennäköisimmin yhä enemmän tulevaisuudessa, siksi ohjelmointia opettavan opettajan (eli jokaisen suomalaisen perusasteen opettajan) olisi hyvä tuntee käytetyimmät ohjelmointikielet ja -keinot edes nimeltä. Yksinkertaistavat esimerkit auttavat niin oppilaita kuin opettajia ymmärtämään, mistä on kysymys. Tästä hyvä esimerkki on olio-ohjelmointi, joka ei välttämättä aukea kovinkaan monille, kun puhutaan aktiivisista tietoalkioista ja niiden suhteista toisiinsa. Näiden sijaan voitaisiin käyttää yksinkertaistavaa esimerkkiä: Jos ajateltaisiin maapallon olevan yksi ohjelma, silloin maapallon asukkaat voisivat olla tuon ohjelman olioita. Kullakin oliolla olisi omia ala-olioitaan, kuten auto, jonka ala-olio olisi puolestaan moottori. Jokaisella oliolla on omat funktionsa,

jotka ratkaisevat ohjelman toiminnan. Asia, mikä tekee olio-ohjelmoinnista erityisen kätevän, on periytyminen (inheritance), jonka ansiota on helppo laajennettavuus. Ohjelmoijan ei tarvitse määritellä asioita useita kertoja, vaan kun ohjelmoija on koodannut esimerkiksi Citroen Xsaran, hän on samalla määritellyt ylemmät luokat, jotka voisivat olla ”henkilöauto”, ”auto”, ”liikkumisväline”. Tästä seuraa se, että kun halutaan koodata VW Golf, on sen tekeminen helppoa, koska molemmat autot periytyvät samoista luokista ja eroavat vain tietyiltä ominaisuuksiltaan.

Kaiken kaikkiaan ohjelmointi tarjoaa mahdollisuuksia monipuoliseen oppimiseen. Eri oppiaineiden sitominen ohjelmointiin voi tapahtua helposti ja oppilaita innostavasti esimerkiksi pelien tekemisen kautta. Yksinkertaisenkin pelin tekeminen vaatii tekijältään monipuolisia ja -alaisia taitoja, kuten suunnittelutaitoja niin pelin tarinan (äidinkieli/englanti) kuin pelin mekaniikan (matematiikka) osalta. Myös kuvankäsittelytaitoja tarvitaan jo heti ensimmäisissä itse tehdyissä graafista ilmaisuja sisältävissä ohjelmissa. Se, että ohjelmoinnin opetteluun tarjoamat mahdollisuudet eivät siis lopu heti kesken, on innostavaa, sillä ohjelmointi antaa hyvät lähtökohdat oppimiselle.

7.1. Keskeiset tutkimustulokset

Tutkielmassa kerätty aineisto tarjoaa runsaasti mielenkiintoisia tuloksia. Tutkielmalle asetettujen tavoitteiden kannalta keskeisimpiä tulokset koskettavat suomalaisten opettajien ohjelmointivalmiuksien kuvaamista sekä asenteita ohjelmoinnin opettamista kohtaan.

Tutkielman tutkimuskyselyllä selvitettyillä vastaajien taustatekijöillä ei ollut kovinkaan suurta vaikutusta vastauksikäyttäytymiseen. Suurimmat erot taustatekijöiden vaikutuksista näkyivät tutkittaessa aiempaa koulutusta ohjelmoinnista. Ne, joilla oli aiempaa opiskelukokemusta ohjelmoinnista, kokivat huomattavasti vähemmän ahdistusta ohjelmoinnin opettamisesta sekä asennoituivat positiivisemmin ohjelmointia kohtaan. Toinen, hieman vähemmän selvästi erottuva ero taustatekijöiden vaikutuksesta oli sukupuolten välillä esiintyvä ero vastauksissa ohjelmoinnin opettamista kohtaan tunnetusta ahdistuneisuudesta, jossa naiset ilmoittivat olevansa hieman ahdistuneempia kuin miesvastaajat.

Vastaajien keskuudessa tunnetuimmat ohjelmointikieliset olivat yleisimmin käytetyt ohjelmointikieliset kuten Java ja HTML/CSS mutta osaamista edellä mainittujen kielten opettamiseen ei ollut juurikaan. Noin yksi kymmenestä vastaajasta kertoi pystyvänsä opettamaan joitakin seuraavista kielistä: Scratch, Scratch Jr., Code.org, ja HTML/CSS. Muiden ohjel-

mointikielien ja -ympäristöjen osaajia oli merkittävästi vähemmän. Uskoa ohjelmoinnin opettamisen osaamiseen vastaajien keskuudessa kuitenkin on, sillä lähes puolet vastaajista ilmoitti pystyvänsä ratkaisemaan ohjelmoinnin opetuksen ongelmia, vaikka paikalla ei olisi ketään ohjeistamassa ja yli 70 % vastaajasta ilmoitti pystyvänsä ratkaisemaan ohjelmoinnin opetuksen ongelmia jos käytettävissä on runsaasti (riittävästi) aikaa. Verrokkitutkimuksen tuloksiin verrattaessa tämän tutkielman suomalaiset vastaajat tunnistivat ohjelmointikieliä heikommin ja valmiudet opettaa ohjelmointia olivat hieman alhaisempia kuin irlantilaisten vastaajien ja huomattavasti alhaisempia kuin pohjoisirlantilaisten vastaajien.

Tutkielman vastaajat suhtautuivat kuitenkin jopa huomattavasti myönteisemmin ohjelmoinnin sisällyttämiseen perusopetukseen kuin irlantilaiset ”potentiaaliset ohjelmoinnin opettajat” ja yhtä myönteisesti kuin pohjoisirlantilaiset tietotekniikan opettajat.

Tutkittaessa opettajien asennetta ohjelmointia kohtaan kaikki tutkimustulokset viittasivat positiiviseen suhtautumiseen (katso 6.4). Kuitenkin kaikilla mittareilla tarkasteltuna suhtautuminen on hieman ”en osaa sanoa” -tason yläpuolella. Verrattaessa Irlannissa tehtyyn tutkimukseen, tässä tutkielmassa tulokset ovat lähempänä neutraalia suhtautumista. Vaikka suurin osa kyselyyn osallistuneista ovat vapaaehtoisesti tulleet ohjelmoinnin opettamista käsittelevälle kurssille, suhtautuminen ohjelmointia kohtaan on yllättävän neutraali. Ohjelmointi on tulossa opetussuunnitelmaan vasta syksyllä 2016, eikä opettajilla ole kovin paljon kokemusta ohjelmoinnin opettamisesta. Nämä näkökulmat huomioon ottaen, voidaan olettaa, etteivät opettajat vielä tiedä aiheesta niin paljon, että voisivat tehdä johtopäätöksiä sen luonteesta. Positiivinen suhtautuminen ja ahdistuneisuuden vähyys kertovat opettajien ottavan ohjelmoinnin ennakkoluulottomasti vastaan. Tätä tukee myös opettajien näkemys ohjelmoinnin tarpeesta peruskoulussa. Teoriaan ja verrokkitutkimukseen pohjautuva korrelaatiotutkimus paljasti sosiaalisen vaikutuksen, vaivannäköodotuksen ja suoriutumiso-dotuksen nopeuttavan opettajien ohjelmoinnin opettamisen aloittamista. Muut tekijät tukevat toisiaan.

7.2. Tutkimuksen rajoitukset sekä luotettavuuden ja eettisyyden loppuarviointi

Tutkimuksen tekemiseen liittyy aina rajoituksia, jotka tulee ottaa huomioon tuloksien tulkinnassa. Tässä tutkielmassa ei ole pyritty antamaan kaiken kattavaa kuvaa suomalaisten opettajien ohjelmointivalmiuksista sekä asenteista ohjelmointia kohtaan. Tutkielman mie-

lenkiinto kohdistui yleiskuvauksen luomiseen, eikä teemaan tässä tutkielmassa ole syvennyt niin syvällisesti, että tutkielman antamaa kuvaa voisi sanoa täydelliseksi.

Tutkielman kyselyyn vastanneet olivat suomalaisia peruskoulun opettajia, jotka olivat ilmoittautuneet verkossa järjestettävälle ohjelmointikurssille, eli vastaajat olivat joko jossain määrin kiinnostuneita ohjelmoinnista tai heitä oli kehoitettu käymään kurssi. Siten, vaikka kysely toteutettiin ennen kurssin alkua, voidaan olettaa että vastaajat edustavat jossain määrin ohjelmoinnin kannalta valveutuneita opettajia. Lisäksi oman rajoituksensa aiheuttaa se, että tutkimusaineisto kerättiin itsearviointiin perustuvalla aineistonkeruumenetelmällä. Tästä saattaa seurata se, että vastaajat antavat itsestään ”sosiaalisesti mielekkäämpiä” vastauksia.

Rajoituksia vertailun osalta aiheuttaa myös verrokkitutkimuksen ”*The programming studio: investigating teachers’ readiness for teaching programming in the Island of Ireland*” vastaajat, jotka eivät asemaltaan vastaa täysin tämän tutkielman vastaajia. Verrokkitutkimuksen pohjoisirlantilaiset vastaajat olivat tietotekniikan opettajia (*teachers of ICT*), kun taas irlantilaiset vastaajat olivat CESI (Computers in Education Society of Ireland) - sähköpostituslistalta tulleita ”potentiaalisia ohjelmoinnin opettajia” (”potential teachers of programming”). Tutkimustulosten vertailu onkin tehty pääosin jälkimmäisen ryhmän kanssa, sillä jälkimmäinen ryhmä on lähempänä tämän tutkielman otosta laadullisesti.

Tutkielman reliabiliteetin kannalta oli olennaista, että tutkimuksen kyselylomake oli samankaltainen verrokkitutkimuksen kanssa, jotta tutkimustulokset olisivat johdonmukaisesti vastaavia, luotettavia ja kestäisivät vertailua. Verrokkitutkimuksen tekijät Cowan, Oldham ja FitzGibbon luovuttivat käyttämänsä kyselylomakkeen sähköpostitse tämän tutkielman tekijöiden käyttöön. Tutkimuksien kyselylomakkeet eivät kuitenkaan olleet täysin yhteneväiset johtuen niin käännöksistä englannista suomeksi kuin myös tämän tutkielman tutkimuskyselyn mahdollistaneen tahon toiveista. Tutkielmalle asetettuja tavoitteita koskevat osiot kyselylomakkeissa olivat kuitenkin käännöksiä mahdollistamissa rajoissa mahdollisimman yhteneväiset. Toinen tutkielman reliabiliteetin kannalta olennainen seikka on se, kuinka johdonmukaisesti ja ristiriidattomasti tutkielman aineisto on kerätty. Kaikki tutkimukseen osallistuneet ovat täyttäneet samanlaisen kyselylomakkeen sähköisellä Webropol-alustalla aikavälillä 8.10.—20.10.2015. Tutkimuskyselyn tarkka aikaraja johtui siitä, että tutkimuskyselyn vastaukset haluttiin ennen kuin kurssin osallistujat pääsivät käsiksi

itse kurssitehtäviin, eli tutkimuskyselyssä haluttiin kartoittaa nimenomaan tilannetta ennen koulutusta.

Tutkielman validiteetin kannalta olennaista on tarkastella sitä, miten hyvin valittu tutkimusmenetelmä soveltuu tutkimusongelman tarkasteluun. Tutkielman tavoitteena oli luoda yleiskuva suomalaisten perusopetuksen opettajien valmiuksista opettaa ohjelmointia sekä selvittää opettajien asenteita ohjelmoinnin opettamista kohtaan. Tutkielma toteutettiin deskriptiivisellä tutkimusotteella ja jo tutkimustulosten yleistettävyyttä sekä havainnoitavan ryhmän, suomalaisten perusopetuksen opettajien lukumäärä, aiheuttavat sen, että tutkimusmenetelmän tulee olla määrällinen.

Tutkielman eettisen tarkastelun kestävydestä vastasi omalta osaltaan tutkimuskyselyn mahdollistaneen koodiaapinen.fi-sivuston koordinaattori Aalto-yliopistosta. Tutkimuskyselyn Webropol-alustalla järjesti ja kyselyssä kerätyn aineiston hallinnoinnista vastaa Aalto-yliopisto. Sama taho vastasi aineiston siistimisestä ennen kuin se luovutettiin tämän tutkielman tekijöille analysointia varten. Siistimisen yhteydessä kaikki viitteet, joilla vastaajan olisi voinut tunnistaa, oli siivottu ja vastaajat olivat aineistossa vain numeroituna 1—317. Tutkielmalle asetettiin National Commissionin (1979) mukaisen eettisyyden arvioinnin kriteerit: *ihmisten kunnioitus*, *hyväntekeväisyys* ja *oikeus*. Eettisen kestävyuden kannalta on olennaista, että tutkittavien täytyy olla sopivia tutkimukseen. Tutkittavat ihmiset edustivat suomalaisia perusopetuksen opettajia, joita tutkielmalle asetettujen tutkimuskysymysten mukaisesti tutkimuksessa tutkittiin. Ihmisten kunnioitus ihmistutkimuksessa näkyi tutkittavien vapaaehtoisuudessa. Tutkimuskyselyyn vastaaminen oli MOOC-kurssin osallistujille täysin vapaaehtoista ja he saattoivat keskeyttää kyselylomakkeen täyttämisen tallentamatta ilman jälkiseuraamuksia. Tutkittavilla oli myös ennen kyselylomakkeen avaamista saatavilla tieto, mistä tutkimuksessa oli kyse, sekä tiedot niin kyselylomakkeen täyttämisen ajallisesta kestosta kuin yksityisyyden suojaamisesta sekä yhteystiedot lisätietojen varalta (National Commission, 1979).

7.3. Tutkimuksen hyöty ja jatkotutkimuksen tarpeet

Tutkielman tuloksista selviää tutkimuskyselyyn vastanneiden koodiaapinen.fi-sivuston järjestämälle ohjelmoinnin MOOC-kurssille ilmoittautuneiden suomalaisten opettajien valmiudet opettaa ohjelmointia sekä näkemykset ohjelmoinnin opetuksesta ennen kurssin alkua, joten tutkielman tuloksia voi jossain määrin yleistää koskemaan kaikkia suomalaisia

perusopetuksen opettajia, jotka tiedostavat ohjelmoinnin opetuksen tulevan osaksi *Perusopetuksen opetussuunnitelman perusteita*.

Tutkielman tekijöiden kannalta mielenkiintoisia jatko- ja lisätutkimuksien mahdollisuuksia ovat esimerkiksi seuraavat kysymykset: Koodiaapinen.fi-sivuston järjestämän MOOC-kurssin koulutuksen vaikuttavuus? Mikä lähestymistapa ohjelmointiin tuottaa parhaita oppimistuloksia milläkin vuosiluokalla? Kannattaako ohjelmoinnin opettaminen aloittaa graafisella käyttöliittymällä vai tekstipohjaisilla ohjelmointialustoilla ja mitkä ovat näiden lähestymistapojen hyödyt ja haitat?

Lähteet

- Alwahaishi, S.;& Snášel, V. (2013). Consumers' Acceptance and Use of Information and Communications. *Journal of Technology Management & Innovation*, 8(2), 61-73.
Haettu 12. 10 2015 osoitteesta <http://www.scielo.cl/pdf/jotmi/v8n2/art05.pdf>
- Beard, R. M. (1971). *Piagetin kehityopsykologia*. (T. Takala, Käänt.) Helsinki: KK:n kirjapaino.
- Binkley, M.;Erstad, O.;Herman, J.;Raizen, S.;Ripley, M.;Miller-Ricci, M.;& Rumble, M. (2012). Defining Twenty-First Century Skills. Teoksessa P. Griffin;E. Care;& B. McGaw (Toim.), *Assessment and Teaching of 21st Century Skills* (ss. 17-66). New York: Springer Dordrecht Heidelberg London New York. doi:10.1007/978-94-007-2324-5
- Cowan, P.;Oldham, E.;& FitzGibbon, A. (2014). The programming studio: investigating teachers' readiness for teaching. Teoksessa A. (. Education);& C. (. Education) (Toim.), *ATEE Annual Conference 2014 - Transitions in teacher education and professional identities* (ss. 299-309). Braga: Institute of Education, University of Minho. Haettu 12. 10 2015 osoitteesta
<https://repositorium.sdum.uminho.pt/handle/1822/36281?locale=en>
- Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*, 13(3), 319-340. doi:10.2307/249008
- Davis, F. D.;Bagozzi, R. P.;& Warshaw, P. R. (1989). User Acceptance of Computer Technology: A Comparison of Two Theoretical Models. *Management Science*, 35(8), 982-1003. Haettu 19. 10 2015 osoitteesta
<http://home.business.utah.edu/actme/7410/DavisBagozzi.pdf>
- Dershem, H. L.;& Jipping, M. J. (1990). *Programming languages: Structures and models*. Belmont, California: Wadsworth Publishing Company.
- Détienne, F. (2002). *Software Design - Cognitive Aspects*. Le Chesnay Cedex: Springer-Verlag London Limited.
- Donaldson, M. (1983). *Miten lapsi ajattelee*. (M.-L. Sakki, Käänt.) Espoo: Weiling + Göös.
- Eid, C.;& Millham, R. (2012). Which Introductory Programming Approach Is Most Suitable For Students: Procedural Or Visual Programming. *American Journal of Business Education*, 5(2), 173-178.

- Flannery, L. P.;& Bers, M. U. (2013). Let's Dance the "Robot Hokey-Pokey!": Children's Programming Approaches and Achievement throughout Early Cognitive Development. *Journal of Research on Technology in Education*, 46(1), 81-101.
- Haaparanta, H. (2008). Tietokoneet perusopetuksen opettajan arkipäivässä: Opettajien työhyvinvoinnin, työuupumuksen ja koulun tietostrategioiden vaikutukset teknologia-asenteeseen. *Tampereen teknillinen yliopisto. Julkaisu 761*. Haettu 10. 11 2015 osoitteesta <http://tinyurl.com/pxkc2s2>
- Hair, J. F.;Anderson, R. E.;Tatham, R. L.;& Black, W. C. (1998). *Multivariate data analysis* (Fifth edition p.). Upper Saddle River, New Jersey: Prentice-Hall, Inc. Haettu 8. 12 2015
- Harsu, M. (2005). *Ohjelmointikielet. Periaatteet, käsitteet, valintaperusteet*. Helsinki: Talentum Media Oy.
- Heikkinen, E.;Haataja, J.;& Halonen, T. (2007). *Laskennallisen tieteen kehittäminen Suomessa. Opetusministeriön työryhmämuistioita ja selvityksiä 2007:23*. Opetusministeriö.
- Horowitz, E. (1983). *Fundamentals of Programming Languages*. New York: Springer-Verlag Berlin-Heidelberg New Yourk.
- Hyvönen, M.;Lappalainen, V.;& Lakanen, A.-J. (2012). *Ohjelmointi 1 C#* (2. Korjattu painos p.). Jyväskylä: Jyväskylän yliopisto, tietotekniikan laitos.
- Järvinen, E.-M. (2001). *Education about and through technology. In search of more appropriate pedagogical approaches to technology education*. Oulu: Oulun yliopisto.
- Kerlinger, F. N. (1993). *Foundations of behavioral research* (Third edition p.). Forth Worth, Philadelphia: Harcourt Brace Jovanovich College Publishers. Haettu 8. 12 2015
- Ketokivi, M. (2009). *Tilastollinen päättely ja tieteellinen argumentointi*. Helsinki: Hakapaino. Haettu 8. 12 2015
- Koodiaapinen. (2. 11 2015). *Koodiaapinen.fi*. (IT-kouluttajat ry) Haettu 2. 11 2015 osoitteesta <http://koodiaapinen.fi/mooc/>
- Kuusela, J. (2000). *Tieteellisen paradigman mukaisen ajattelun kehittyminen peruskoulussa - Kahden interventiomenetelmän vertaileva tutkimus peruskoulun kuudesluokkalaisilla*. Helsinki: Helsingin yliopiston opettajankoulutuslaitos.

- Lavonen, J. M.; Meisalo, V. P.; & Lattu, M. (2001). Problem Solving with an Icon Oriented Programming Tool: A Case Study in Technology Education. *Journal of Technology Education*, 12(2), 21-34.
- Lazar, J.; Feng, J. H.; & Hochheiser, H. (2010). *Research Methods in Human-Computer Interaction*. Chichester, West Sussex: John Wiley & Sons Ltd.
- Lehtinen, T. (Toim.). (1. Tammikuu 2014). *Helsingin sanomat, kotimaan uutiset*. Haettu 3. 8 2014 osoitteesta <http://www.hs.fi/kotimaa/a1390279526604>
- Lindh, M. (2014). Käsityötiede, käsityökasvatus vai teknologiakasvatus? - Teknisen työn taustateoriasta. *Tekninen Opettaja*, 1, 6-7. Haettu 2. 12 2015 osoitteesta <http://www.tekninenopettaja.net/docs/Kasityotiede,%20kasityokasvatus%20vai%20teknologiakasvatus.pdf>
- Liu, Y.; Li, H.; & Carlsson, C. (2010). Factors driving the adoption of m-learning: An empirical study. *Computers & Education*, 55(3), 1211-1219. Haettu 19. 10 2015 osoitteesta <http://www.sciencedirect.com/science/article/pii/S0360131510001508>
- Louden, K. C. (2003). *Programming Languages. Principles and Practice*. San Jose: Thomson Learning.
- Majaranta, P. (2002). Ohjelmointi on helppoa - lapsikin sen osaa. Teoksessa P. Hietala; & S. Ovaska (Toim.), *Lasten Käyttöliittymät*. Tampere: Tampereen yliopiston tietojenkäsittelytieteiden laitos.
- Malmberg, N. (Toim.). (16. 10 2013). *Yleisradio*. Haettu 21. 9 2014 osoitteesta http://yle.fi/uutiset/suomen_koulujen_it-opetus_jordanian_tasolla_uutta_mallia_haetaan_virosta/6885099
- Mannila, L.; Peltomäki, M.; & Salakoski, T. (2006). What About a Simple Language? Analyzing the Difficulties in Learning to Program. Teoksessa L. Mannila (Grandell), *Teaching Mathematics and Programming: New Approaches with Empirical Evaluation* (ss. 211-227). Turku: TUCS Dissertations.
- Metsämuuronen, J. (2004). *Pienten aineistojen analyysi - Parametrittomien menetelmien perusteet ihmistieteissä*. Jyväskylä: Gummerus Kirjapaino Oy.
- Metsämuuronen, J. (2005). *Tutkimuksen tekemisen perusteet ihmistieteissä*. Jyväskylä: Gummerus Kirjapaino Oy.
- Michigan Tech. (9. 11 2015). *Research: Informed Consent Details*. Noudettu osoitteesta Michigan tech: <http://www.mtu.edu/research/administration/integrity-compliance/review-boards/human-subjects/informed-consent.html>

- Mitchell, J. C. (2004). *Concepts in programming languages*. Stanford University: Cambridge University Press.
- Mittermeir, R. T. (2013). Algorithmics for Preschoolers - A Contradiction? *Creative Education*, 4(9), 557-562.
- National Commission. (1979). *The Belmont Report: Ethical Principles and Guidelines for the Protection of Human*. The National Commission for the Protection of Human Subjects. Haettu 3. 11 2015 osoitteesta
<http://www.hhs.gov/ohrp/humansubjects/guidance/belmont.html>
- Neittaanmäki, P.; Lehto, M.; & Kankaanranta, M. (2014). *Kohti laskennallisen ajattelun osaamista*. Informaatioteknologian tiedekunta. Jyväskylä: Jyväskylän yliopisto.
- Opetushallitus. (2014a). *Perusopetuksen opetussuunnitelman perusteet 2014*. Haettu 2. 5 2015 osoitteesta
http://www.oph.fi/download/163777_perusopetuksen_opetussuunnitelman_perusteet_2014.pdf
- Opetushallitus. (2014b). *Opettajat Suomessa 2013. Koulutuksen seurantaraportit 2014:8*. Tampere. Haettu 16. 11 2015 osoitteesta
http://www.oph.fi/download/156282_opettajat_suomessa_2013.pdf
- Paananen, J. (2005). *Tietotekniikan peruskirja*. Jyväskylä: Docendo.
- Pekkala, L. (2002). Animoitu ohjelmointiympäristö lapsille. Teoksessa P. Hietala; & S. Ovaska (Toim.), *Lasten käyttöliittymät* (ss. 174-183). Tampere: Tampereen yliopiston tietojenkäsittelytieteiden laitos.
- Pietikäinen, J. (2013). *Pro gradu -tutkimus: Teknologian hyväksyminen ja käyttö kotihoidossa*. Aalto-yliopisto, Tieto- ja palvelutalouden laitos. Helsinki: Aalto-yliopisto. Haettu 14. 10 2015 osoitteesta
http://epub.lib.aalto.fi/fi/ethesis/pdf/13127/hse_ethesis_13127.pdf
- Sajjanhar, A.; & Faulkner, J. (2014). Exploring Second Life as a Learning Environment for Computer Programming. *Creative Education*, 5(1), 53-62.
doi:<http://dx.doi.org/10.4236/ce.2014.51010>
- Sempere, A. (2005). Animatronics, Children and Computation. *Educational Technology & Society*, 8(4), 11-21.
- Sola, H. (2002). Ohjelmointia ilman koodausta. Teoksessa P. Hietala; & S. Ovaska (Toim.), *Lasten käyttöliittymät* (ss. 184-194). Tampere: Tampereen yliopiston tietojenkäsittelytieteiden laitos.

- Töttö, P. (2000). *Pirullisen positivismin paluu. Laadullisen ja määrällisen tarkastelua.* Tampere: Osuuskunta vastapaino.
- Venkatesh, V.;& Davis, F. D. (2000). A Theoretical Extension of the Technology Acceptance Model: Four Longitudinal Field Studies . *Management Science* , 46(2), 186-204. doi:<http://dx.doi.org/10.1287/mnsc.46.2.186.11926>
- Venkatesh, V.;Morris, M. G.;Davis, G. B.;& Davis, F. D. (September 2003). User Acceptance of Information Technology: Toward a Unified View. *MIS Quarterly*, 27(3), 425-478. Haettu 12. 10 2015 osoitteesta <https://csdl-techreports.googlecode.com/svn/trunk/techreports/2005/05-06/doc/Venkatesh2003.pdf>
- White, G. L.;& Sivitanides, M. P. (2002). A Theory of the Relationships between Cognitive Requirements of Computer Programming Languages and Programmer's Cognitive Characteristics. *Journal of Information Systems Education*, 13(1), 59-66.

Liitteet

Liite 1: Kyselylomake ohjeistuksineen. Varsinainen kysely on toteutettu Webropol-alustalla tämän lomakkeen ohjeiden pohjalta

1. Sukupuoli

Radionapit, joissa: 2=Mies, 1=Nainen

2. Ikä

Pudotusvalikko, jossa vaihtoehdot:

-15

16-19

20-24

25-29

30-34

35-39

40-44

45-50

50-54

55-59

60-64

65-69

70+

3. Kuinka monta vuotta olet toiminut opettajana?

Tyhjä kenttä: Syötä numero

4. Mitä vuosiluokkaa opetat tällä hetkellä?

Pudotusvalikko, jossa:

1-2

3-6

7-9

muu

en opeta peruskoulussa

5. Tulin kurssille, koska:

Pudotusvalikko, jossa:

1 Työnantajani suositteli vahvasti

2 Työtoverini suositteli kurssia

3 Löysin kurssin itse ja kiinnostuin

4 Jokin muu syy

6. Kerro osaamisesi seuraavista ohjelmointikielistä (Voit valita useamman vaihtoehdon jokaiselta riviltä ja voit lisätä luetteloon enintään 3 ohjelmointikieltä, joita pidät relevantteina)

	1. En ole koskaan kuullut tästä kielestä	2. Olen kuullut tästä kielestä	3. Olen käyttänyt tätä kieltä viimeisen 3 vuoden aikana	4. Harrastan ohjelmointia tällä kielellä	5. Voisin opettaa op- pilaille tätä kieltä	6. Voisin opettaa toi- sille opetta- jille tätä kieltä
Scratch						
Scratch Jr.						

Racket						
Code.org						
Turtle Roy						
Hopscotch						
Lego RXT, NXT tai EV3						
JavaScript						
Python						
C/C++						
Java						
Processing						
C#						
Logo						
AppInventor						
Pascal						
Cobol						
Greenfoot						
Picaxe basic						
Basic (Visual, Cool, tms.)						

HTML, CSS						
-----------	--	--	--	--	--	--

7. Onko sinulla loppututkintoa ohjelmoinnin opinnoista?

Radionapit, joissa

0 = Ei

1 = Ei - mutta olen opiskellut itsenäisesti

2 = Kyllä

Kysymyksiin 8 – 30 ja 32 – 36: kysymyksen jälkeen radionapit, joiden yläpuolella on vaihtoehdot vasemmalta oikealle:

1 = Täysin eri mieltä

2 = Osittain eri mieltä

3 = En osaa sanoa

4 = Osittain samaa mieltä

5 = Täysin samaa mieltä

Kysymyksiin 31 ja 37 - 39 on vastausvaihtoehtojen ohjeistus kysymyksen kohdalla

Suoriutumisodotukset ohjelmoinnin opettamisesta.

8. Ohjelmoinnin opettaminen on hyödyllistä työni kannalta
9. Ohjelmoinnin opettaminen auttaa minua suoriutumaan työstäni nopeammin
10. Ohjelmoinnin opettaminen tekee opetuksestani tehokkaampaa
11. Jos opetan ohjelmointia, onnistun työssäni paremmin
12. Ohjelmoinnin opettamisen osaaminen monipuolistaa opettajuuttani
13. Ohjelmoinnin opettamisen osaaminen parantaa työllistymistäni.

Vaivannäköodotukset

14. Ohjelmoinnin opetukseni tulee olemaan selkeää ja ymmärrettävää

- 15. Minusta tulee helposti taitava ohjelmoinnin opettaja
- 16. Minusta ohjelmoinnin opettaminen on helppoa
- 17. Ohjelmoinnin opettamisen oppiminen on minulle helppoa

Asenteet ohjelmoinnin käyttöä kohtaan

- 18. Ohjelmoinnin opettamista tarvitaan peruskoulussa
- 19. Ohjelmoinnin opetus tekee työstäni mielenkiintoisempaa
- 20. Ohjelmoinnin opettaminen on hauskaa
- 21. Pidän ohjelmoinnin opettamisesta

Sosiaalinen vaikutus ohjelmoinnin opettamiseen

- 22. Minulle tärkeiden ihmisten mielestä minun tulisi opettaa ohjelmointia
- 23. Käyttäytymiseeni vaikuttavien ihmisten mielestä minun tulisi opettaa ohjelmointia
- 24. Tämän kurssin opettajat ovat avuksi ohjelmoinnin opettamisen oppimisessa
- 25. Työpaikallani suhtaudutaan positiivisesti ohjelmoinnin opetukseen

Helpottavat olosuhteet

- 26. Työpaikallani on tarvittavat välineet ohjelmoinnin opetukseen
- 27. En voi hyödyntää ohjelmointia muussa opetuksessa
- 28. Minulla on riittävät tiedot ohjelmoinnin opettamiseen
- 29. Saan tarvittaessa tukea työyhteisöstäni, jos minulla on ongelmia ohjelmoinnin opettamisessa
- 30. Minulla voi olla vaikeuksia selittää, miksi ohjelmointia kannattaa tai ei kannata opettaa oppilaille.

Kyvykkyys

Kysymykseen 31 tarvitaan valintalaatikot, eli vastaaja voi valita halutessaan vaikka kaikki vastausvaihtoehdot

- 31. Valitse kaikki sinulle sopivat väittämät jatkoksi lauseelle: ”Voin ratkaista ohjelmoinnin opetuksen ongelmia...”

- a. ”vaikka paikalla ei olisi ketään ohjeistamassa”
- b. ”jos voin soittaa apua ongelmatapauksissa”
- c. ”jos minulla on runsaasti aikaa perehtyä ongelmaan”
- d. ”jos minulla on käytettävissäni apua tai tukea”

32. Pyrin käyttämään uutta teknologiaa opetuksessani aina kun se on mahdollista.

Ahdistus

- 33. Epäroin opettaa ohjelmointia, koska saatan opettaa sitä väärin**
- 34. Minua pelottaa ajatella, että voin menettää paljon tietoja jos painan väärää nappia**
- 35. Tunnen ahdistusta ohjelmoinnin opettamista kohtaan**
- 36. Ohjelmoinnin opettaminen on minulle vierasta ja pelottavaa**

Aikomuksesta opettaa ohjelmointia

- 37. Olen halukas esittelemään ohjelmointia koulullani**

Kysymyksiin 37 – 39 tarvitaan tyhjä kenttä <n> paikalle, johon vastaaja voi kirjoittaa vastauksen numeroina

- 38. Suunnitelmissani on opettaa ohjelmointia tulevien (syötä numero) <n> kuukausien aikana**
- 39. Voisin kuvitella opettavani ohjelmointia seuraavien (syötä numero) <n> kuukausien aikana**
- 40. Tarkoitukseni on, että tulen opettamaan ohjelmointia seuraavien (syötä numero) <n> kuukausien aikana**